UCONN | UNIVERSITY OF CONNECTICUT

Final Report

F4U Corsair Flight Simulator
Spring 2014

CT Corsair Senior Design Team

Zachary Mosch, Randy Bertrand,
Amanda Sweat, David Tartaglino
uconn.ee.corsair@gmail.com

Lauren Bradley, Arthur Podkowiak,
Michael Turner
uconn.me.corsair@gmail.com

Project Advisors

Rajeev Bansal
ITE, 463
(860) 486-3410
rajeev@engr.uconn.edu

Stephen Stagon
Engineering II, 311
(860) 486-5088
stephen.stagon@engr.uconn.edu

SPONSORED BY CT CORSAIR

CONNECTICUT CORSAIR

TABLE OF CONTENTS

1. Summary and Overview……3
2. Introduction……3
2.1. Sponsor Background……3
2.2. Simulator Background……3
2.3. Problem Statement……4
2.4. Design Goals……4
3. Design Details……5
3.1. Design Deliverables……5
3.2. Overview and Description of Proposed Solution and Contingency Design……6
3.2.1. Clarifications on Induction vs. Servo Systems……7
3.3. Induction Motor Rebuild: the How and Why……10
3.4. Electrical Design Modifications……10
4. Project Specifications……11
4.1. Simulator Motion Definitions and Requirements……11
4.2. Motion and Environmental Specifications……12
4.3. Final Motor Specifications……13
4.4. Parametric Model Development……13
4.5. Prepar3D Compatibility……13
5. Kinematic Analysis for Motor Selection……14
5.1. Theory……14
5.2. Free Body Analysis: Spring Coefficient……14
5.3. Free Body Analysis: Torque Requirements……16
5.4. Torque Requirements……21
5.5. Angular Velocity Calculations……21
5.6. Horsepower Requirements……22
6. Prototype……22
6.1. Purpose of Prototype……22
6.2. Construction and Functionality……23
7. Motors and Drives Control……24
7.1. DS2110 Digital Controller Commissioning Software……24
7.2. Moog DS2110 Servo Drive Setup Procedures……27
7.3. DS2110 Commutation Offset……28
7.4. Servo Motor and Drive System Components……29
7.5. Motor Programming and Communications……29
8. Scissor Arm Design……31
8.1. Scissor Arm Overview……31
8.2. Purpose of Redesign……31
8.3. Scissor Arm Theory and Free Body Analysis……31
8.4. Upper Scissor Arm: Finite Element Analysis……32
8.4.1. Model Selection……32
8.4.2. Analysis Criteria……33
8.4.3. Analysis Results……35
8.5. Lower Scissor Arm Design……35
8.5.1. Parametric Model Development……35

8.5.2. Finite Element Analysis: Model Selection and Criteria……36
8.5.3. Analysis Results……36
8.5.4. Tensile Test Validation……37
8.5.5. Finite Element Analysis Results……38
9. Motor Mount Design……39
9.1. Motor Mount Overview……39
9.2. Parametric Model Development……39
9.3. Finite Element Analysis……40
9.3.1. Model Selection……40
9.3.2. Analysis Criteria……41
9.3.3. Analysis Results……41
9.3.4. Motor Mount Analysis Conclusions……42
10. Conclusion……42
10.1. Summary of Project Accomplishments……42
10.2. Project Integration Challenges……46
10.3. Future Recommendations……47

A1. Nomenclature……49
A2. References……51
A3. Supplementary Analysis Data……52
A4. Final Wiring Diagram……54

1 Summary and Overview

Connecticut Corsair has sponsored an interdisciplinary team of senior engineering students to restore a damaged Gyro IPT flight simulator, developed by Environmental Tectonics Corporation (ETC), with obsolete components and software to working condition. The ultimate goal of this project is to have the simulator respond to user input via standard airplane controls such that the simulator mimics the flight patterns of the F4U-4 Corsair aircraft. The goal for this year's senior design team has been the restoration of three-axis motion. To accomplish motion restoration, the team needed to replace the lower scissor arms and successfully select, calibrate and install system appropriate motors, drives and gearboxes.

2 Introduction

2.1 Sponsor Background

Connecticut Corsair is a non-profit organization, founded in 1991, dedicated to restoring an F4U-4 Corsair to flying condition[12]. The F4U Corsair was used primarily in World War II and the Korean War as a fighter aircraft. The wings on the aircraft folded upward for storage aboard aircraft carriers and the pilot was positioned over the wings in a domed cockpit, allowing for a full view during flight[27]. Craig McBurney, the founder of Connecticut Corsair, is sponsoring this project. Craig has helped to maintain vintage aircrafts at more than 400 aviation museums throughout the country, and is exceptionally knowledgeable in the flight characteristics of the F4U-4 Corsair aircraft.

Figure 1. F4U-4 Corsair with folded wings[27]

Figure 2. F4U-4 Corsair in flight[22]

2.2 Simulator Background

Connecticut Corsair received a donated flight simulator system from Environmental Tectonics Corporation® (ETC), a company based out of Pennsylvania specializing in aviation and space training equipment for both military and civil applications[2]. The original simulator was a first generation ETC Gyro IPT™ simulator, shown in Figure 3**Error! Reference source not found.**, which mimicked the flight patterns of large jets. The simulator has been kept in storage for an extended period of time, and has had various components stripped to support commercial

models. The simulator provided to the team is equipped with three induction motors and accompanying VFDs intended to control the pitch, roll, heave, and vestibular movements of the simulator. These motors are not capable of running the simulator for an extended time period due to their tendency to overheat. To combat this problem ETC installed cooling fans on the motors which act as a quick fix. The encoder feedback elements of these induction motors are quite old and no adequate customer support is offered by their original manufacturer. A new means of feedback is necessary for integrating the induction motors with the control software.

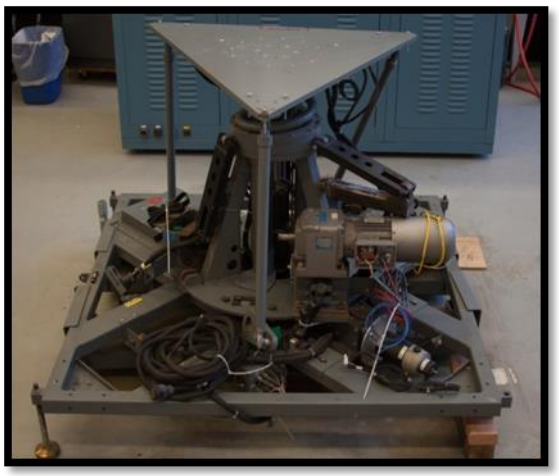


Figure 3. Donated Gyro IPT™ Simulator Base

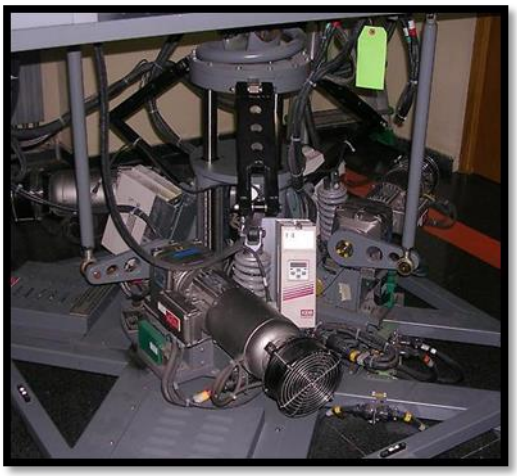


Figure 4. Original Gyro IPT™ Simulator Base

2.3 Problem Statement

Connecticut Corsair is an entirely volunteer organization relying solely on donations in order to complete the restoration. Multiple interdisciplinary senior design teams will be working on simulator renovations from year to year, their composition varying on the demands of the project. The ultimate goal of this project is to have the simulator respond to user input via standard airplane controls such that the simulator mimics the feel and flight patterns of a WWII F4U-4 Corsair aircraft. Connecticut Corsair intends to use the simulator for promotional purposes to raise support for the restoration of the original aircraft[12].

As of September 2013, the Corsair Simulator restoration project is in its second phase in a multi-phase plan as proposed by the initial capstone senior design team. Last year's work in Phase I centered on system analysis and research. This year's senior design team has been asked to restore three-axis motion to the simulator's base. The Gyro IPT™ flight simulator model has a triangular base with three actuating pushrods, three scissor-arm attachments and a central supportive spring.

2.4 Design Goals

This year's design team is focused exclusively on the simulator's base; responsible for total system motion. The simulator cockpit has been removed to emphasize the base as the priority and to make safety paramount as mounting takes place. The planned culmination of this phase will result in the base moving up and down, side to side, and backwards and forwards in response to a user input. A fourth motor is used for system "spin" or yaw, and is outside the scope of this project.

To accomplish motion restoration, the team needed to replace the lower scissor arms and successfully select, calibrate and install system appropriate motors, drives and gearboxes. Accompanying requirements included the construction of auxiliary control circuitry, the mapping of IO requirements from hardware to software and an exploration of the proposed Prepar3D simulation software.

3 Design Details

3.1 Design Deliverables

This year's Phase II design team consists of senior undergraduate mechanical, electrical and computer engineering students. The deliverables for the project are appropriately divided amongst the disciplines depending on their areas of expertise.

The first deliverable of Phase II is the *establishment of the simulator's torque and electrical requirements.*

The second deliverable follows directly from the first deliverable: *the selection of motors and gearboxes which best fit the established design constraints*. It is critical that the motors selected do not interfere with any moving simulator components. Motor selection then facilitates the selection of gearboxes, motor drives, control systems and commissioning software.

The third deliverable is a *prototype used as a test-bed for control code development.*

The fourth deliverable of the project was a *new motor mount design*. This was used to attach the new gearbox and motor to the simulator base.

The fifth deliverable was the *repair, installation and control of all simulator motors***.** This deliverable involved the replacement of induction hardware pieces, an understanding of high level analog servo drives, appropriate control circuitry and controlled programming via an appropriate microcontroller.

The sixth deliverable was the *incorporation of induction motor feedback and limit switches as safety features***.** Feedback was obtained via potentiometer voltage readings. These were incorporated onto encoder wheels. Upper and lower limit switches were mounted on the maximum bounds of the moving cam, and wired to provide a flag to the controlling program that direction of the cam must immediately be adjusted.

The seventh deliverable of Phase II is a *redesign of the simulator's lower scissor arms.* These arms attach to the bottom of the upper triangular base piece connected to the simulator base. They act as dampers against the simulator's heaving. The redesign is necessary due to an over-engineered and incorrectly sized model created by the previous Phase I senior design team.

The eighth deliverable of the project was a *working SolidWorks assembly of the whole simulator base.*

The ninth deliverable was the addition of *user-controlled three-axis motion using adjustable joystick inputs.*

The final deliverable was a *study and preparation of the Prepar3D flight simulation software.* This software was outside the scope of our project goals and this deliverable is intended to assist future design teams.

3.2 Overview and Description of Proposed Solution and Contingency Design

Restoration of the simulator's 3-axis movement involves the interfacing of a user inputs to a microcontroller. This communicates to three separate motor drives via analog communication which in turn drive three rotational braking motors via an amplified signal. The motor shafts are connected to gearboxes which modify the output torque and speed. The real position is read by feedback devices. These devices communicate back to the microcontroller.

To select an effective gearbox/drive/motor combination, three companies were contacted; Moog, Yaskawa, and Bosch Rexroth. Yaskawa motors are of exceptional quality but are expensive compared to other motor brands. Moog has a branch completely dedicated to flight simulation. They are also one of the only companies that sell both servo motors and drives. It is for this reason Moog was selected as the motor supplier.

The team developed primary and backup motor design solutions. The primary solution utilizes a Moog servo motor and analog drive system alongside the two original Nord braking induction motors and KEB Combivert VFDs. Figure 5 illustrates this design scheme. The backup design utilizes three Nord braking induction motors, two KEB VFDs and one Yaskawa VFD. During the servo drive system development phase, this alternative contingency plan was created in the event failure occurred with the servo drive system. The Yaskawa VFD in question was borrowed for a short time from a generous faculty member at the University of Connecticut.

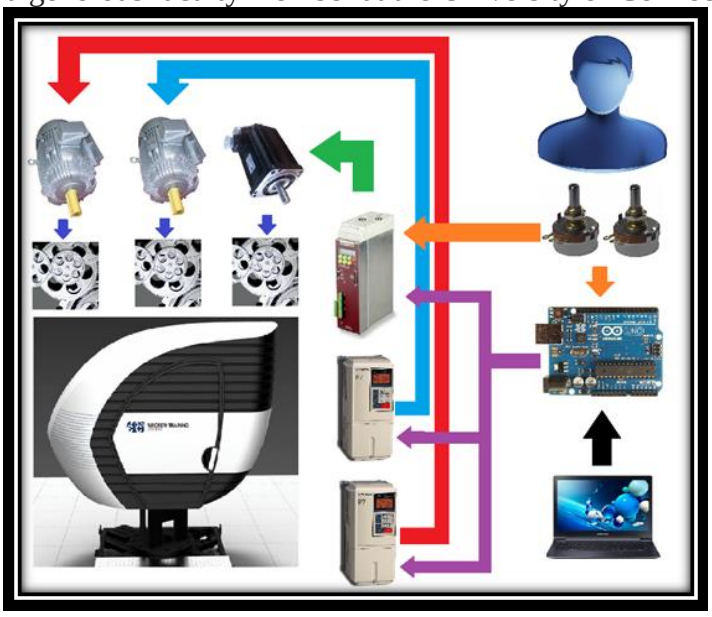


Figure 5: Visual Representation of Project Goals

The following points outline the general workflow of the primary servo/induction installation and design process.

- Repair Induction motors
- Write, test and demonstrate system control code program for induction motors
- Servo drive factory fault clearance
- Factory reset via high-level access code
- Commissioning software installation
- Commissioning software comprehension
- Commissioning software parameter refinement
- Servo actuator tuning
- Demonstrate one time actuator shaft control with microcontroller
- Establish automatic data uploading
- Recode servo encoder feedback adjustments
- Install and incorporate limit switch flags to code
- Install and incorporate variable voltage potentiometer feedback for induction motors
- Component mounting and soldering
- Final kinematic model testing

The following points outline the alternative three induction motor design plan.

- Repair Induction motors
- Clean and fix up third induction motor: stator, housing and gaskets
- Wire and install Yaskawa VFD
- Modify relay switching circuit with extra relays and wirings
- Write, test and demonstrate system control code program for one induction motor
- Microcontroller code duplication
- Install and incorporate limit switch flags to code
- Install and incorporate variable voltage potentiometer feedback for induction motors
- Component mounting and soldering
- Final kinematic model testing

3.2.1 Clarifications on Induction vs. Servo Systems

Induction motors are controlled using VFDs to vary input signal frequencies. This changes the motor shaft's output speed and torque. Induction motors require separate encoders on the output shaft to get position feedback. Thus, position control is more difficult using induction motors. The current Nord motor position encoder is shown in Figure 6.

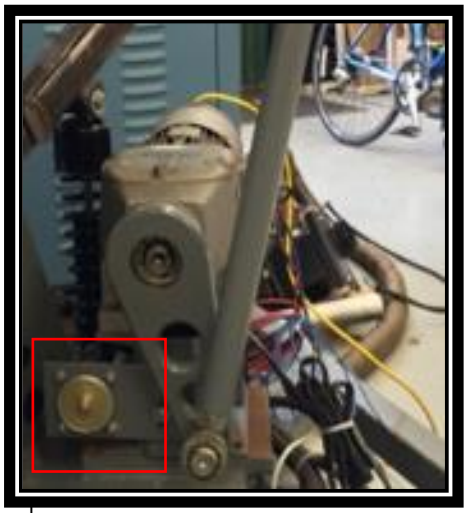


Figure 6. Induction Motor Encoder

Servo drives receive command signals from a controller. They amplify the signal and transmit current to an actuator to produce a proportional motion. The command signal typically represents velocity but can also represent torques or positions. Unlike induction, servos have an integrated feedback sensor which reports the shaft's status to the drive. The drive compares actual status with the commanded status. It then alters the applied signal pulse width to correct for deviations.

Servo motors are the best choice for this application due to their integrated encoder, minimal size, and power efficiency. Although induction motors are half the price on average of a servo motor, the sponsor asked the team to disregard cost when performing motor analysis. An extensive comparison of both induction and servo motors and drives can be found on the following page in Table 1.

	Criteria	Servo Advantages	Servo Disadvantages	Induction Advantages	Induction Disadvantages
Motor	Encoder	▪ Encoder integrated ▪ Accurate position control	▪ Program required for gearbox ratio correction		▪ Program required for gearbox ratio correction ▪ Encoder separate ▪ Not accurate
	Size	▪ Less weight ▪ Smaller		▪ Similar in size to original motors	▪ Heavy ▪ Up to 50% larger
	Price		▪ Approx. $6000/motor ▪ Expensive: high quality materials & internal feedback	▪ Approx. $3300/motor ▪ Less expensive due to external encoder	
	Efficiency	▪ 90% efficient ▪ Magnet rotors lose less power between stator and rotor			▪ 60%-70% efficient ▪ High power loss due to motor materials
	Heat Waste	▪ Low operating temperature ▪ Low heat production			▪ High operating temperature ▪ High heat production
	Operating Current	▪ Low current draw ▪ No magnetizing current required, rotor is a permanent magnet			▪ High current draw ▪ Magnetizing current to make magnetic field for rotation
	Prototyping	▪ Hobby servo motors cheap to prototype ▪ Development libraries available			▪ Difficult to prototype: small induction motors not readily available
Drives	Software	▪ Manufacturer software included ▪ Reduced programming complexity	▪ Some drives don't come with manufacturer software		
	Controllers	▪ Microcontrollers may be an option depending on brand	▪ Drives require controller; hard to code ▪ Price greater than microprocessor	▪ Microprocessors can be used with VFDs; easy to code ▪ Cheaper than controllers	
	Price		▪ Expensive	▪ VFDs are less expensive	
	Compatibility		▪ Motor/drive must come from same manufacturer	▪ Any VFD works with any motor	
	Control	▪ Positional control is easy using servo drives due to built-in encoder		▪ Speed control is easy using VFDs. Speed proportional to input frequency	▪ Complicated position control

Table 1. Pros and Cons of Induction and Servo Motors and Drives

3.3. Induction Motor Rebuild: the How and Why

The Nord induction motors and gearbox accompanying the simulator are the original pieces of hardware installed while the simulator was being developed at ETC. Both natural and unnatural wear and tear have resulted in damages to the motor housings and several internal pieces. The gearboxes were also in pieces, and one of three VFDs only worked in turning the induction motor in the positive direction. As two induction motors are a part of the proposed design, some work was required to get them operational.

Last year's team purchased two non-braking induction motors for their proof of concept design. Good fortune (or planning) had it that these two non-braking motors were of the same build; with the exception of the brakes of course. The internal stators of the non-braking motors were removed and installed in the braking motors. After several attempts to repair the dings and cracks in the housings, the non-braking housings were then placed on the braking motors.

3.4 Electrical Design Modifications

Upper and Lower Limit switches were added to the design as a safety feature. The switches are mounted above and below the maximum degree limits of cam rotation. Figure 7a and 7b show two perspectives of an upper limit switch.

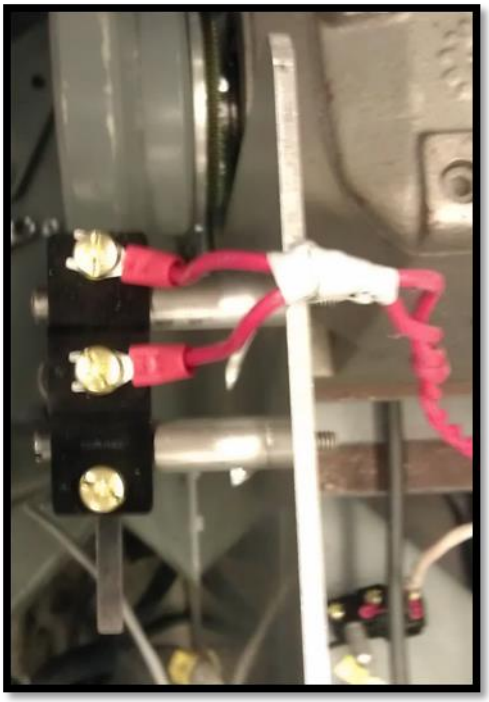


Figure 7a. (top view of upper limit switch)

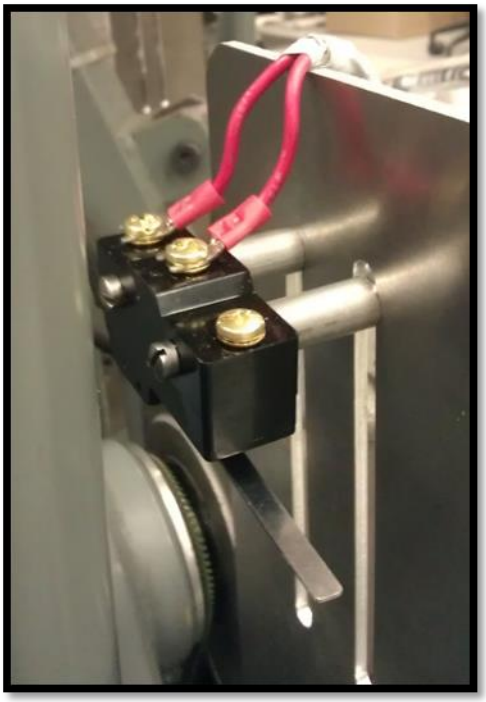


Figure 7b. (side view of upper limit switch.)

Two 1kΩ resistance potentiometers are being used instead of the joystick originally intended for use. The joystick's internal variable resistances of 1MΩ were too high and not enough current was drawn through the control circuitry.

The original simulator's induction motor encoders were replaced by 1MΩ pots. These pots were fastened to the encoder gear as shown in Figure 8. Their signals are read and processed directly

by the microcontroller. Their purpose is to protect the machine from exceeding its limits by providing positional analog voltage information from the two induction motors.

Figure 8. Induction Motor Encoder Feedback via Potentiometer

This custom feedback device was selected for use on the simulator for several reasons. First, the encoders are very old and no support exists for their model and datasheets are nearly impossible to find. The encoder manufacturers provided minimal assistance in understanding their workings. Second, while testing motor shaft positions, the encoder output a 16-bit value of all ones or all zeros and nothing in between. Code developed by both the electrical and computer engineering team members produced these same results. Finally, future work on the simulator will require the installation of two more servo motors which will render the induction motors and all parts pertaining to them obsolete. Thus, potentiometer feedback was selected as the more viable option.

4 Project Specifications

4.1 Simulator Motion Definitions and Requirements

The Gyro IPT flight simulator has a triangular base plate where the cockpit rests. There are three actuating pushrods, three scissor arm attachments and a central universal joint surrounded by a supportive spring. As previously stated, the simulator had three induction motors to control pitch, roll, heave and vestibular movements of the simulator.

Pitch is defined as the rotation of the simulator about its center x axis and is controlled by the front drive motor. Pitch is used to simulate climbing, diving and acceleration. Roll is defined as the rotation of the simulator about its center y-axis and is controlled by the two side drive motors. Rotation of the simulator about its y-axis allows for the simulation of turns and rolls. Heave is defined as vertical excursions in the z-direction from the home position. It is controlled by all three drive motors working in unison to provide lift. Heave simulates runway roughness, landings and turbulence. Vestibular movements are defined as movements which stimulate the inner ear balance system and can be used to give the user a sensation of acceleration.

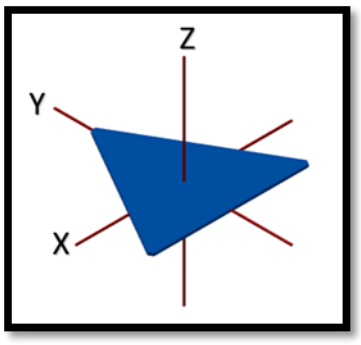


Figure 9. Axes of motion for simulator platform

Yaw is the rotation about the vertical z-axis. This maneuver is to be controlled by the ‘spin’ motor and is outside the scope of this project phase.

Pitch and roll are dependent on the center of the simulator platform having a fixed pivot point. The pivot point is obtained by connecting the center of the simulator platform to a spring and universal joint combination. The universal joint is then connected to the platform and the guide rod ring which eliminates translational linear movement of the simulator platform along the x and y directions but will allow for rotational movement about the x and y axes.

The motors utilize a gear box and cam pushrod system to induce platform motion. The gearbox output shaft rotates the cam. The cam translates the rotational motion of the gearbox shaft into linear motion. Translation from one form of motion to the next happens through a ball pivot joint which allows the pushrod to rotate perpendicular to the cam. The opposite end of the pushrod connects to a knuckle joint attached to the platform. The knuckle joint allows the pushrod to move in the x-y plane. Freedom of the pushrod to rotate about the knuckle joint and cam is critical to allow the pushrod to apply force to the platform regardless of orientation.

4.2 Motion and Environmental Specifications

The range of motion for the Corsair simulator is expected to correspond with the original Gyro IPT™ simulator characteristics as found in Table 2.

Displacement Type	Range of Motion	Speed	Acceleration
Pitch	+/- 25 deg/sec	0-25 deg/sec	0.5-75 deg/sec^2
Roll	+/- 30 deg/sec	0-25 deg/sec	0.5-75 deg/sec^2
Yaw	360 degree continuous	0-150 deg/sec	0.5-15 deg/sec^2
Heave	+/- 10 cm	30 cm/sec	90 cm/sec^2
Sway	+/- 10 cm	20 deg/sec	60 deg/sec^2
Surge	+/- 10 cm	20 deg/sec	60 deg/sec^2

Table 2. Gyro IPT™ Simulator Range of Motion[18]

The original simulator runs on three phase power. The environmental and physical specifications for simulator operation are listed in 3. It is important to note that although future designs may change, most operating specifications from the original Gyro IPT™ will remain the same in order to preserve system integrity.

Parameter	Requirement
Voltage	220-240 Volts AC
Frequency	50/60 Hz
Phase	Three Phase
Nominal Current Rating	10 Amps
Surge Current Rating	13 Amps
Protective Device Rating	16 Amps
Expected Operating Temp Range	+13°C to +35°C
Expected Humidity Range	10% to 80% non-condensing

Table 3. Operating Conditions for Gyro IPT™ Simulator[18]

4.3 Final Motor Specifications

Although final motor specifications are inherently dependent on gearbox ratio selection, the calculations performed in Section 5 dictate the systems overall speed and torque requirements. Table 4 contrasts the original Gyro IPT™ specs with the new Corsair simulator specs.

Criteria	Gyro IPT™	Corsair Simulator
Motor Type	Induction	Servo
Speed	26 rpm	85 rpm
Torque Output	1212 in-lbs	3500 in-lbs
Power	1.5 hp	5 hp

Table 4. Original Gyro IPT™ Specifications vs. Corsair Simulator Specifications

4.4 Parametric Model Development

To perform a finite element analysis and derive the necessary torque equations, an accurate model of the simulator base was created. A parametric model was developed in Solidworks by replicating every component in the base critical to the structure as separate files. Each part was measured using calipers with tolerances of ±.001 in. These files were mated and given relations in an assembly drawing so valid point-to-point dimensions could be found. The accuracy of the upper scissor arms and spring measurements were imperative to the model's success as those parts mate to the new lower scissor arms. The upper scissor arm model was exported into ABAQUS to perform the finite element analysis.

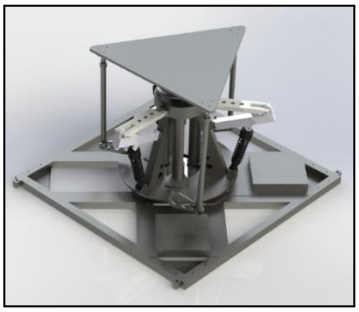


Figure 10. Rendering of Solidworks Parametric Model [25]

4.5 Prepar3D Compatibility

Prepar3D is a flight simulation software developed by Lockheed Martin. Future design teams will work to integrate the simulator with a license of Prepar3D Academic v 1.4.4747.0. It is used for both entertainment and training purposes. The user is provided with an immersive cockpit experience and is able to load real-life airports and environments. There is also a custom mission design option.

Figure 11. Screenshot of Prepar3D Flight Simulation [20]

Software development can be done using the SimConnect Developer SDK. Software add-ons are flexible and can be written in C, C++ C#.net or VB.net. Developers are free to add gauge processing, replace events, record and monitor flights, create custom missions, manipulate the weather, enable hardware with Prepar3D and control additional views and/or AI aircraft. The key to communicating with the simulator hardware is the development of a custom client which communicates with the Prepar3D software in run time. This can be done by modifying source code and was outside the scope of Phase II.

An F4U-4 Corsair Plane Model was purchased from A2A; a software add-on company. The file is from their "Aircraft Factory" line and includes F4U-4 A, B, C and D models. By installing the provided executable, a 3D Corsair airplane is loaded into the program with skins, dashboards and expected flight characteristics.

5 Kinematic Analysis for Motor Selection

5.1 Theory

The original prototype's induction motors are 1hp. These motors were replaced in later commercial models as they tended to overheat with prolonged use[13]. The overheating occurs when motor's proper duty cycle is exceeded. When the duty cycle is exceeded, the motor does not have enough time to cool the between cycles[19]. The induction motors obtained by the Phase I team are 0.5hp induction motors with a 90° offset on the output shaft and were also non-braking, which rendered them not an option for the Phase II design. To obtain reasonable torque requirements, a free body analysis was performed on the mechanical system which includes a derivation of the central spring constant, the forces experienced by the cam's arm and the required output angular velocity.

5.2 Free Body Analysis: Spring Coefficient

Due to the presence of the central universal joint, the spring cannot act in lateral z-direction compression or tension. For the derivation of the spring constant, it is assumed that the only motion the experienced by the spring is a rotational motion about the central pivot point[17]. It is assumed that the spring constant can be derived by assuming the right side of the pivot point provides upward force, while the left side of the spring applies a downward, tilting force which restores the platform to a neutral flat position. The free body diagram of the platform is shown in Figure 13. The central spring and universal joint is depicted in red and the platform in teal.
Knowledge from statics[14] can be used to describe the moment about the pivot point. This involves the spring and tilting forces, which can be summed and set equal to zero.

Figure 12. Central universal joint and spring motion

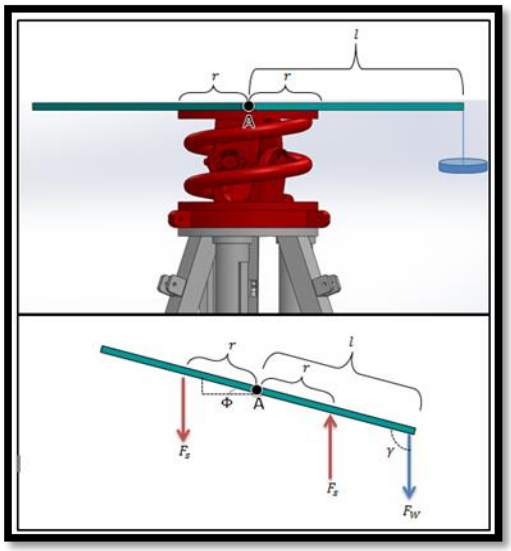


Figure 13. Free Body Diagram for Calculating Spring Coefficient

$$+\circlearrowleft \sum_{Ma} F_s r \cos\Phi + F_s r \cos\Phi - F_w l \cos\Phi = 0 \qquad \textit{(Equation 1)}$$

The spring force is defined as:

$$F_s = ky \qquad \textit{(Equation 2a)}$$

Where k represents the spring constant in N/m and y is the linear displacement of one side of the spring. This equation, becomes *Equation 2b* due to the shared spring force between the three arms.

$$F_s = \frac{ky}{3} \qquad \textit{(Equation 2b)}$$

The free body diagram reveals the relation between the angle Φ, and the displacement of the simulator in the y-direction. This relation is expressed in:

$$r \sin\Phi = y \qquad \textit{(Equation 3)}$$

After substituting *Equation 2b* and *Equation 3* into *Equation 1* the following equation is obtained.

$$k = \frac{3F_w l}{2r^2 \sin\Phi} \qquad \textit{(Equation 4)}$$

With the relation between spring constant and platform force derived, an experiment was conducted to determine the spring constant. An angle finder was place on top of the platform in line with the pivot point and pushrod location. The angle finder has a tolerance of 0.5º, and as a result the experiment is limited by this accuracy. A chain was attached to the platform at the pushrod location and weights ranging from 5 to 320 lbs were hung off of the edge (depicted in Figure 12). The weights represented varying amounts of pulling force that a pushrod could exhibit on the platform. The weights have been calibrated by a certified metrologist to ensure the weights were accurate within ±.001 lbs.

Each weight and corresponding displacement was recorded and plotted to verify that the spring constant is a linear relationship. The spring constant was found to be 189000 N/m. As seen in Figure 14, the weights between 110 to 320 pounds were used in the derivation due to measurement tool limitations. The range of weights between 5 to 105 pounds did not result in a noticeable displacement due to the large spring constant size. Raw data from the experiment can be found in Table 1-1 of Appendix 3: Supplementary Analysis Data.

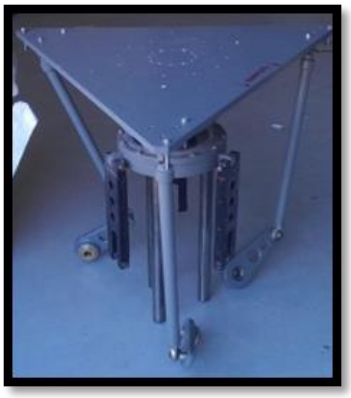


Figure 13. Components Contributing to Static Weight of Simulator

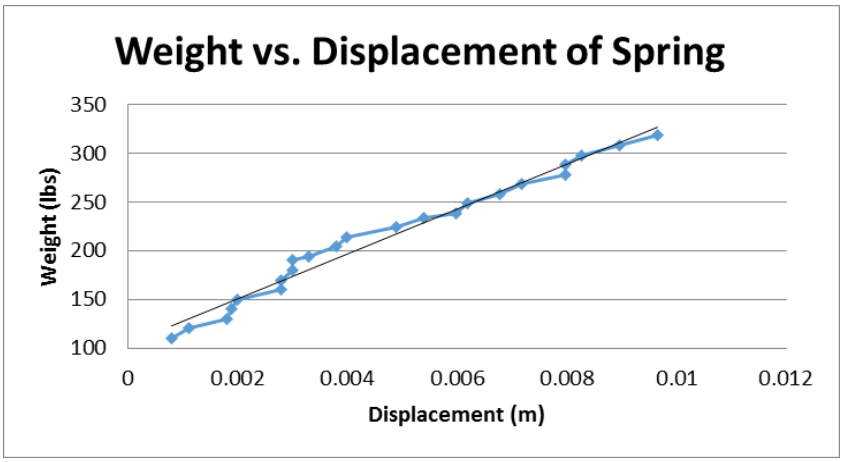


Figure 14. Plot of Weight vs. Displacement for Spring Coefficient

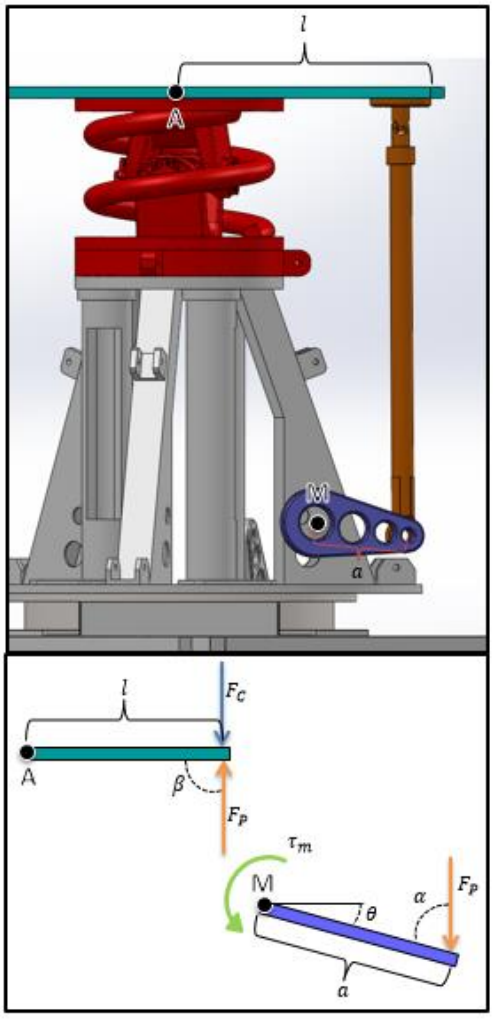


Figure 15. Free body diagram for torque analysis based on vertical lift

5.3 Free Body Analysis: Torque Requirements

To determine the motor torque requirement, it was necessary to find the total static simulator weight the motors have to overcome. This weight includes pushrods, the central spring, the universal joint and the platform. These components are shown in Figure 13. Using a platform scale, this weight was determined to be 240lbs.

Included in the simulator weight was an estimate of the cabin weight. This was broken into two portions: the pilot weight and the structure weight. The maximum pilot was estimated to be 250lbs as recommended in ETC's original Gyro IPT user manual[18]. The cabin structure was estimated around 750lbs, which includes external features and installed instrumentation. This estimate was determined by the Phase I senior design team based on the original simulator cockpit components. This is considered to be an overestimate as the new components are planned to have a lighter weight.

To represent the maximum output torque provided by the motors to move the simulator in the z-direction, the kinematic equations for the simulator's vertical lift and heave were derived. The lift sequence was accepted to be the distance the platform travels from its rest position to the highest point the assembly can be lifted. To

develop the equations, the free body diagrams were drawn to include the cam, pushrods, platform and central spring joint. The diagram depicts the analysis of one motor, as the loading for each motor is assumed to be identical.

In this diagram, Point **A** is the platform pivot point, Point **M** is the motor output shaft, F_c is the weight of the platform assembly, F_p is the force of the pushrod, l is the length from Point **A** to the pushrod, a is the length of the cam, α is the angle between the pushrod and the cam, β is the angle between the platform and the pushrod, θ is the angle between the horizontal plane and the cam and τ_m is the motor torque.

The moments were summed about Point **A** and Point **M**.

$$+\circlearrowleft \sum_{Ma} \quad F_p l \sin\beta - F_c l = 0 \qquad \textit{(Equation 5)}$$

$$+\circlearrowleft \sum_{Mm} \quad \tau_m - F_p a \sin\alpha = 0 \qquad \textit{(Equation 6)}$$

Solving *Equation 5* for the push rod force and substituting the results into *Equation 6*, an equation for the vertical lift torque requirement is developed.

$$\tau_m = \frac{F_c a \sin\alpha}{\sin\beta} \qquad \textit{(Equation 7)}$$

The value F_c does not represent 1240lbs, but rather 413lbs which is one third of the total weight. This assumption was made since the weight will be equally distributed over the platform. Equal distribution of weight can be assumed because the pilot will be centered on the simulator platform.

It is important to note the relationship depicted between α, β, and θ by *Equation 7*. The relationship between these three angles is not linear due to the non-linear motion of both the cam and simulator platform. To better understand this relationship, a 3D model of the simulator was created in Autodesk Inventor.

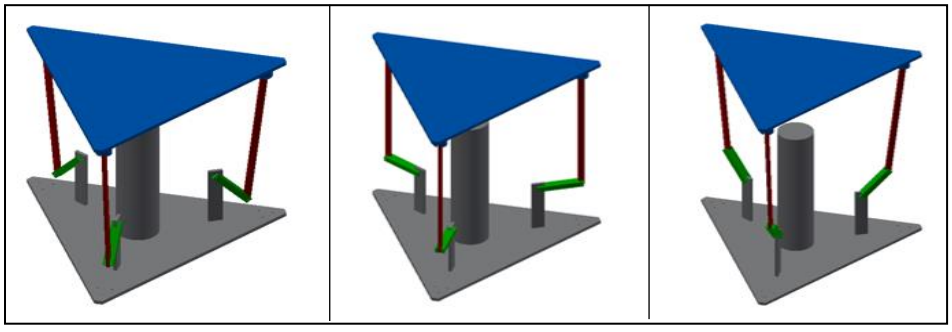


Figure 16. Autodesk Inventor simulating vertical lift: θ less than zero (pictured left), θ equals zero (pictured center), and θ greater than zero (pictured right). This simulated the relationship between the cam (green), pushrods (red), and platform (blue)

The cam represented in green, the pushrod is red, and the platform blue. Using this model, the cam was driven by θ in the positive and negative z-direction from -100° to +90° in increments of 10°. This range was selected based on the range of motion the cam can reach. At each increment the angles α and β were measured in the 3D model. By plotting the relationship between these angles in Microsoft Excel, and determining the line of best fit, two equations were established to find angles α and β as a function of θ shown in Figure 17. These equations were used in a MATLAB code to calculate the torque based on any θ and *Equation 7*.

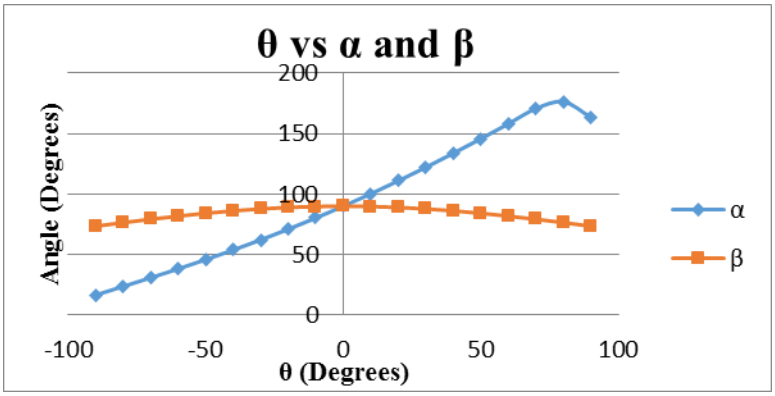


Figure 17. Plot of relationships between θ vs. α and θ vs. β

Figure18. shows the torque curve for the θ range between -100° and 90° and shows that maximum torque occurs when θ=0, when the cam and platform are perpendicular to the pushrod. Achieving maximum torque at this point is valid because all of the pushrod force is perpendicular to the cam which creates the largest force on the cam. After analysis the torque required to lift the simulator in a vertical z-direction was found to be 2894.3 in-lbs.

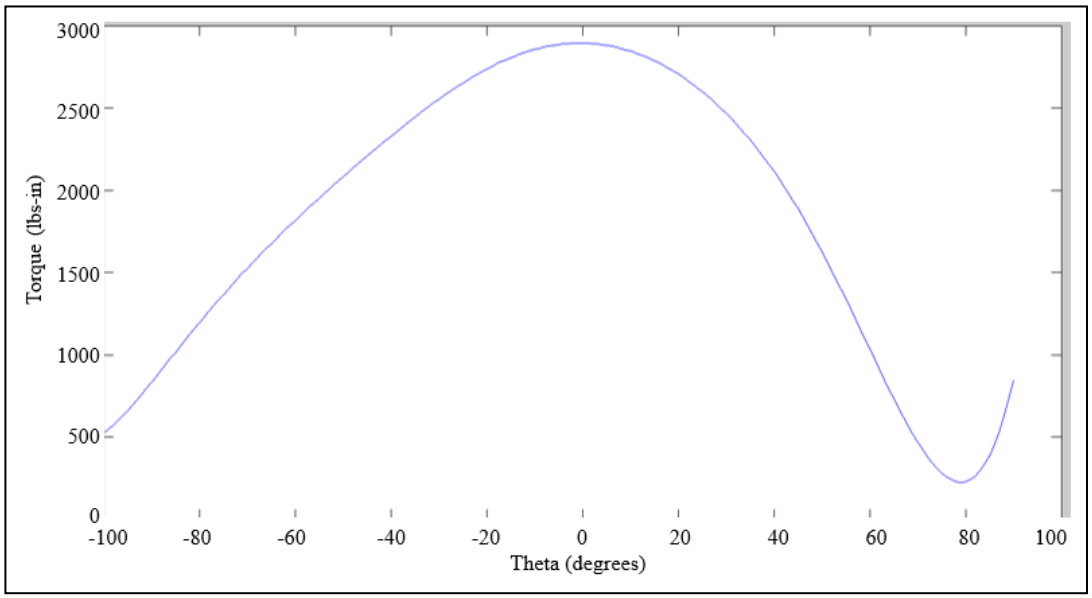


Figure 18. Plot generated using MATLAB showing torque calculated at each θ. The maximum torque occurs at θ = 0

The next simulation analyzed was the simulator pitch and roll. For this analysis, it is still assumed that all three motors provide equal force to induce. The free body diagrams were derived and are shown in Figure 19.

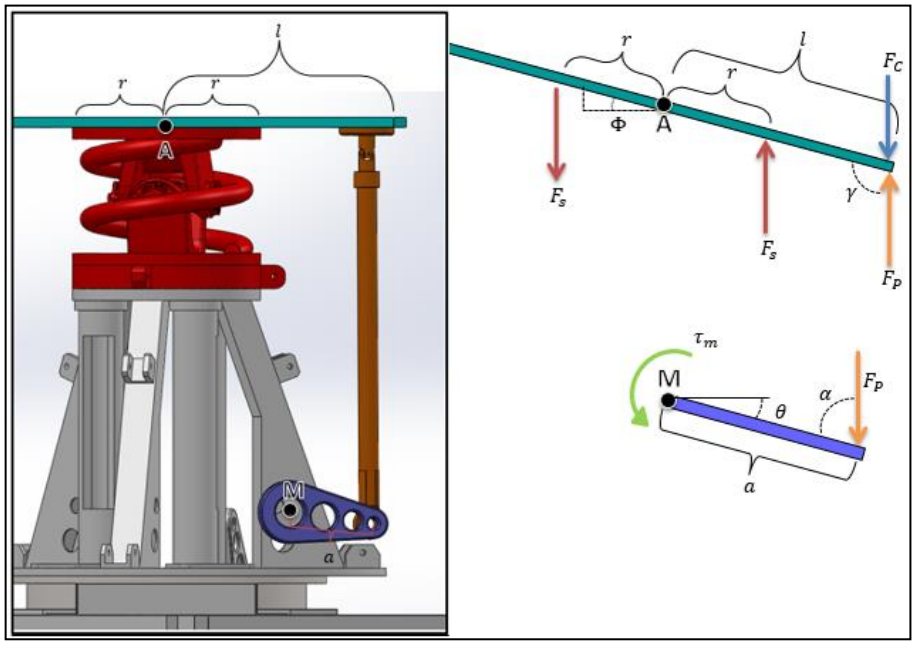


Figure 19. Free body diagram for torque analysis based on the pitch and roll of the simulator

In this diagram, Point **A** is the platform pivot point, Point **M** is the motor output shaft, F_c is the weight of the platform assembly, F_p is the force of the pushrod, l is the length from Point **A** to the pushrod, a is the length of the cam, α is the angle between the pushrod and the cam, γ is the angle between the platform and the pushrod, Φ is the angle between the horizontal plane and the platform, θ is the angle between the horizontal plane and the cam and τ_m is the motor torque.

By summing the moments about Point **A** and Point **M** the following two equations are established.

$$+\circlearrowleft \sum_{Ma} \quad F_s r\cos\Phi + F_s r\cos\Phi - lF_c\cos\Phi + F_p l\sin\gamma = 0 \qquad \textit{(Equation 8)}$$

$$+\circlearrowleft \sum_{Mm} \quad \tau_m - F_p a\sin\alpha = 0 \qquad \textit{(Equation 9)}$$

Equation 8 was solved for the pushrod force, and substituted into *Equation 9*. The spring force was substituted as derived in *Equation 2b* and *Equation 3*. This substitution yielded an equation for the torque necessary to pitch or roll the simulator.

$$\tau_m = a\sin\alpha\left[\frac{F_c\cos\Phi}{\sin\gamma} - \frac{2kr^2\cos\Phi\sin\Phi}{3l\sin\gamma}\right] \qquad \textit{(Equation 10)}$$

As with the previous calculation, it is important to note that the angles do not vary linearly with each other. To understand the relation between α, γ, Φ, and θ a 3D model was created of just the cam, pushrod, and platform. Figure 21 shows this model with green representing the cam, red representing the pushrod, and blue representing the platform. The model again used 10° intervals between -100° and 90° for θ. For each θ, the angles α, γ, and Φ were recorded. Each of these angles were plotted against θ in Microsoft Excel, and curve fitting was done to establish relationships between θ and the other three angles. This relationship is shown in Figure 20.

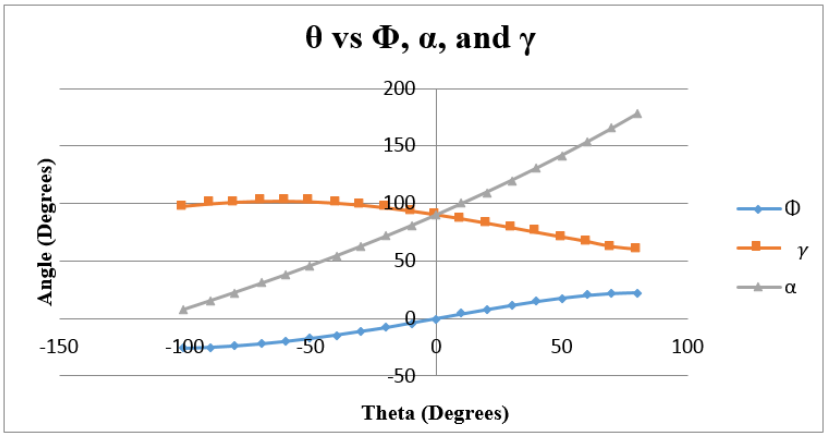


Figure 20. Plot of relationships between θ vs. Φ, θ vs. α and θ vs. γ

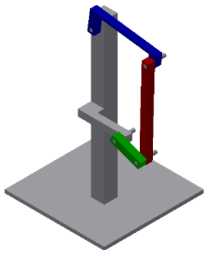


Figure 21. Autodesk Inventor model of cam (green), pushrod (red), and platform (blue) relationship

Additional MATLAB code was written to determine the torque based on any given θ. Figure 22 shows the torque curve for the range of θ from -100° and 90° which shows that maximum torque occurs when θ=27° with a value of 3484.1 in-lbs. The max torque does not occur when θ=0° due to the addition of the spring force. The further the simulator pitches and rolls, the higher the torque the motor receives from the cam.

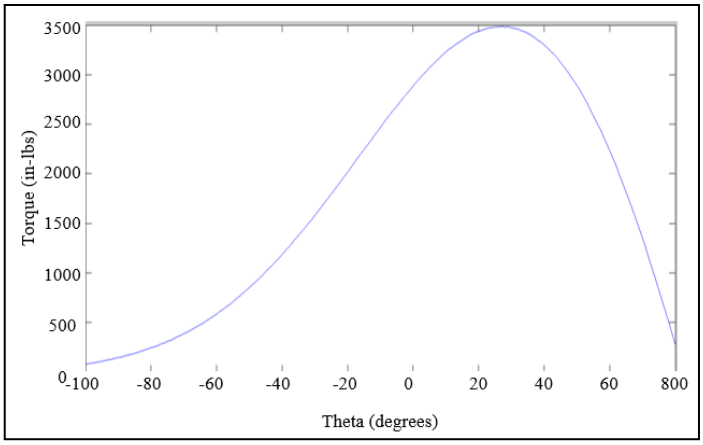


Figure 22. Plot generated using MATLAB shows torque calculated at each θ. The max torque occurs at θ=27°

5.4 Torque Requirements

After evaluation of the three main simulator movements, the total torque requirements are 3484.1 in-lbs and 2894.3 in-lbs. The larger torque value was selected and rounded up to 3500 in-lbs.

The validity of this number was assessed by two means. First, when comparing the two calculated torque values, they were within 600 in-lbs of each other. These numbers analytically are within a reasonable range from each other as the only difference in the assessments is the addition of the spring force. The second validity check consisted of correspondence with a professional in the field. Our professional reference, Charles Bartel, works at Moog Motors[4]. This company develops motors for flight simulation equipment and has a department devoted specifically to flight simulators. Mr. Bartel has designed a flight simulator similar to the ETC simulator and utilized a motor and gearbox combination which yielded 3500 in-lbs per pushrod. This number is almost identical to the calculated torque requirements.

A safety factor is included in the selection of the torque requirement. The previous simulator mimicked the movements of a jet simulator, and the torque requirements from the existing motors proved adequate for the purposes of the project. Although the minimum requirements for the motor motion was performed by the existing motor specifications, the renovated simulator will have an increased acceptable torque range to handle the quicker movements of a Corsair aircraft.

5.5 Angular Velocity Calculations

Utilizing the previous studies, two angular velocities were generated using the ETC recommended speeds from Table . ETC recommends 30 cm/sec of heave which was combined with the change of height of the simulator platform as calculated in the first torque derivation. To calculate the degrees per second necessary to achieve the desired lift rate the following equation was derived:

$$\frac{\Delta\theta}{\Delta z}\frac{degrees}{cm}\cdot 30\frac{cm}{s}=\frac{degrees}{s} \qquad \textit{(Equation 11)}$$

This equation states that for every 10° θ travels, the platform traversed a certain distance in the z-direction. The value is multiplied by the maximum rate at which the simulator traverses the z-axis. The calculated value for required velocity based on heave criteria was found to be 59.5 RPM.

The required angular velocity for pitch and roll could not be found in a similar manner as the relationship is not linear like the heave motion. Instead, the roll velocity for a Corsair was researched and found to be 81°/sec[18]. Using the analysis from the second torque calculations, the following equation was developed for the pitch and roll angular velocity:

$$\frac{\Delta\theta}{\Delta\phi}\frac{degrees}{degrees}\cdot 81\frac{degrees}{s}=\frac{degrees}{s} \qquad \textit{(Equation 12)}$$

This equation states that for every 10° θ travels, the platform will have rolled a distance represented by ΔΦ. The equation was multiplied by the required roll rate of 81°/sec. Roll rates based on θ are displayed in Table 1-3 in **Error! Reference source not found.**. The greatest angular velocity was chosen from Table 1-2 (also in Appendix 3) to be 85 RPM. As a result, the simulator requires a gearbox-motor combination that operates at 85 RPM.

5.6 Horsepower Requirements

To calculate the power requirements of the motors the following calculations were performed using system specifications from Table 3.

To calculate the necessary motor input speed:

$$rpm = \frac{120f}{\#\ of\ poles} = \frac{120 \cdot 60\ Hz}{4} = 1800\ rpm \qquad \textit{(Equation 13)}^{26}$$

The gear ratio was calculated using:

$$R = \frac{\omega_{in}}{\omega_{out}} = \frac{1800\ rpm}{85\ rpm} = 21.1 \qquad \textit{(Equation 14)}^{26}$$

Where ω is speed. Input torque was evaluated using the calculated torque output and the gear ratio.

$$\tau_{in} = \frac{\tau_{out}}{R} = \frac{3500\ in \cdot lbs}{21.1} = 165.9\ in \cdot lbs \cdot \frac{1\ ft}{12\ in} = 13.8\ ft \cdot lbs \qquad \textit{(Equation 15)}$$

Finally, the horsepower requirement was calculated using:

$$P = \frac{\tau_{in} \cdot \omega_{in}}{5252} = \frac{(13.8\ ft \cdot lbs) \cdot (1800\ rpm)}{5252} = 4.7\ hp \qquad \textit{(Equation 16)}^{26}$$

Since 4.7hp is not a standard motor size, the number was rounded up to 5hp.

6. Prototype

6.1 Purpose of Prototype

The team constructed of a small scale, inexpensive 3-axis motion servo motor prototype using hobby servo motors. Its purpose was to serve as a test bed for the Arduino sketch code used for motor control, and to gain experience in programming using positional feedback control in a low-risk environment. These small servo motors communicated with the microcontroller using simple analog IO and provided feedback from their internal encoders. The Moog motor drive used in the final design also communicates with the microcontroller using this analog IO and encoder feedback. Thus, while waiting for the shipment of the servo motor this prototype provided the team with a means to get a head start on the code development.

6.2 Construction and Functionality

The first prototype iteration used an Arduino Uno and LEDs to symbolized motor speed with light intensity. The second prototype implemented hobby motors. The microcontroller used was an Arduino Uno. Arduino technology has numerous open source libraries available, many of which are servo motor specific. Log files were created to take the potentiometer inputs and communicate the data to the controller. The Arduino Uno specifications are outlined in

.

Parameter	Specification
Microcontroller Type	ATmega328
Operating Voltage	5V
Recommended Input Voltage Range	7-12V
Critical Input Voltage Range	6-20V
Total Digital I/O Pins	14
PWM Output Pins	6/14 of Total I/O Pins
Analog Input Pins	6
DC Current for I/O Pins	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB, 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Table 1. Arduino Uno specifications[3]

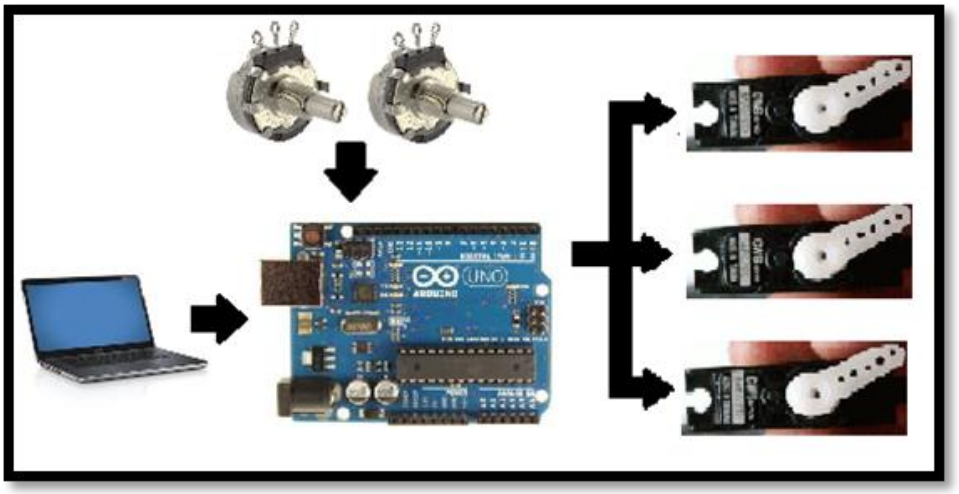


Figure 23. Prototype diagram

Since the current and voltage requirements of the motors was small, the microcontroller IO pins were wired directly to the servo cables. Two potentiometers represented the two dimensional movement of a joystick. The computer provided power to the board and also uploaded the developed programs.

The functionality of this prototype contributed to the proof of concept of the simulator programming by providing a means to enforce device data flow. Variable voltage inputs, representing joystick position, and encoder feedback position inputs were fed to a bounded, two dimensional mapping function which then resulted in motor position output. The range of motion was represented using 226 bit locations in both the x and y directions. Pairs of these values conform to the expected motion of the platform, as seen in Figure 24.

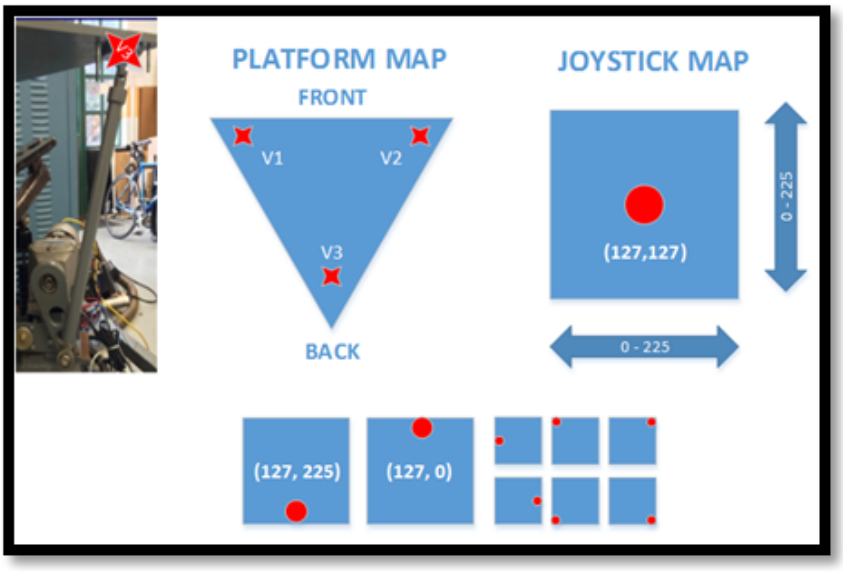


Figure 24. Joystick Mapping Example

7 Motors and Drives Control

7.1 DS2110 Digital Controller Commissioning Software[30]

The DS2110 Drive is controlled using custom commissioning software also produced by Moog. After setup, the software provides the operator with a GUI used for enabling and control purposes. The GUI establishes communication with the DS2110 Drive using serial communications via an RS232 serial null modem protocol.

The following procedure was then used to configure the software for the purposes of running the actuator/drive system.

The first step was to establish what kind of feedback device the program was to expect. By default, the program expects resolver formatted feedback. After selecting an encoder as the feedback device, the drive was restarted and the program refreshed. The next step was to set up the program to recognize the actuator model being used. Because this software is typically used for linear actuators, a linear actuator model with the same parameters as the rotational actuator was chosen from the actuator library with the help of a Moog representative.

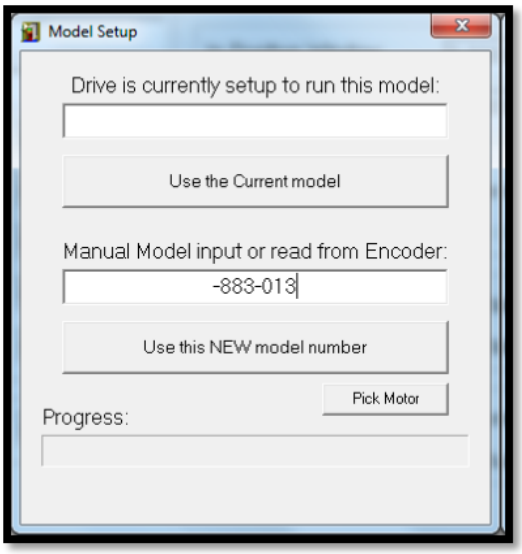


Figure 25. Model Setup Window for Model Selection

Rotational Actuator (Actual)	Linear Actuator (Same Parameters)
G405-1034A	G415-800A PTC

Table 6. Actuator Model Numbers

After the motor model is selected, the software parameters could then be set and adjusted using its internal menus which include *System Setup*, *I/O Setup*, and *Motion*. By default, the actuator was then launched in Commissioning Mode. This mode limits output speed and torque to 10% of the possible maximum. By entering the *Diagnostics Window*, this percentage was adjusted to 20%. It is here that the PIV gains of the controller were tuned and other parameters including fault time outs were modified. See Figure 26 for these values.

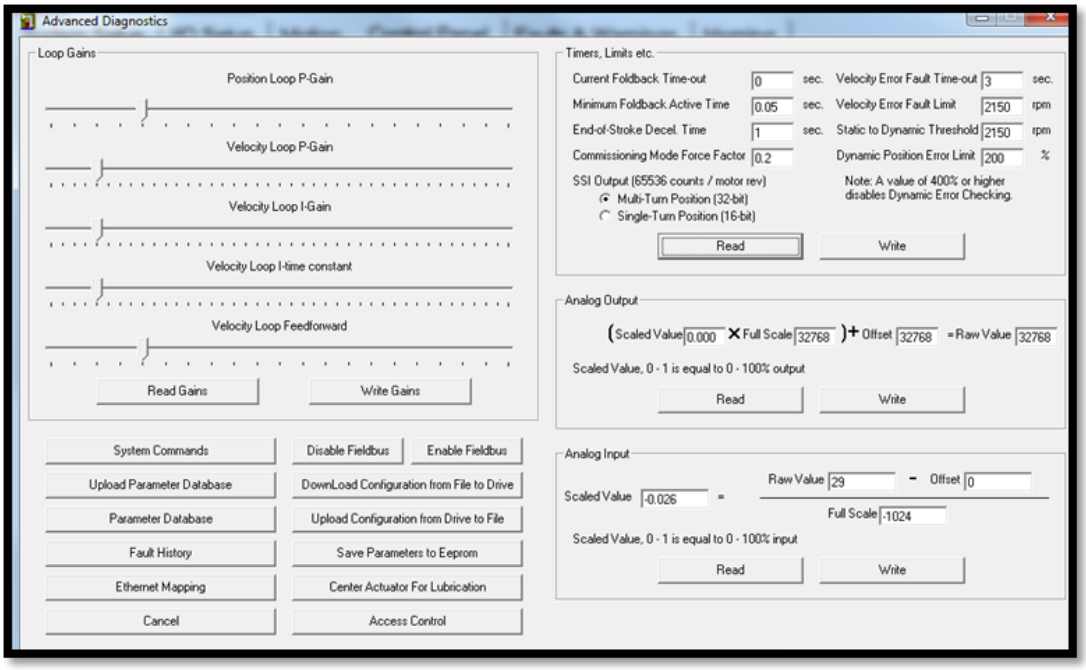


Figure 26. Diagnostics Setup Window

Finally, the drive is controlled using the GUI's Control Panel. This panel is represented in Figure 27.

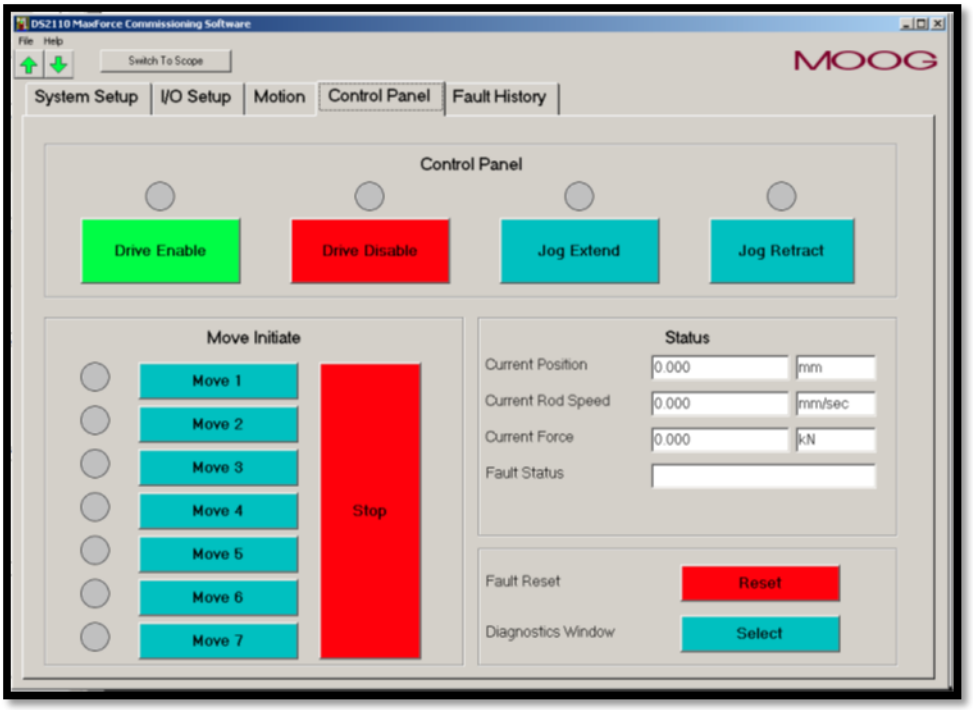


Figure 27. Control Panel

The *Drive Enable* is a software drive enable input, which enables the functions on the software's screen. For actual operation, a hardware enable input at the drive terminal is required for operation. When operating normally, the hardware input enables both the hardware and the software. The *Drive Disable* then disables any and all software inputs.

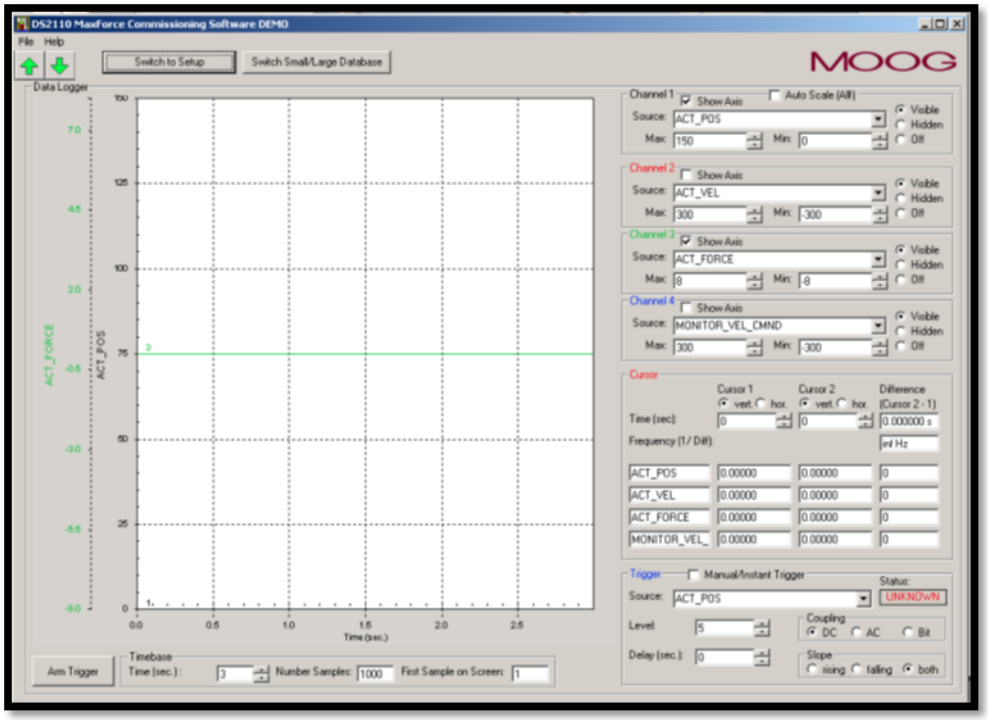
Jogging is used to align or position the drive's actuator shaft for connection purposes or maintenance. The *Jog Extend* and *Jog Retract* buttons are used to utilize the drive's jogging function.

The *Move 1-7* buttons are preprogrammed moves as defined by the operator and will be initiated if and only if the *Drive Enable* is high. Clicking the *Stop* button stops any of these movements that are in progress.

Current Position and *Current Rod Speed* indicates the actuator's current (absolute) shaft position and rod speed. The *Current Force* indicates a +/- (10-15)% error estimate of the force being generated by the actuator shaft. Factors including but not limited to efficiencies and tolerances affect this estimate.

The *Fault Status* bar indicates if any specific fault conditions occur. Likewise, the *Fault Reset* button is used to reset any faults that occur.

The software also provides a useful *Scope Function* which was used to observe motor behaviors.



Function 28. Scope Function

The drive is controlled by the Arduino UNO microcontroller via a 9-pin analog male connector. The drive continuously reads inputs from the microcontroller and communicates these commands to the actuating shaft. It then provides adjustments by reading encoder feedback data and sending this information to the microcontroller for processing.

7.2 Moog DS2110 Servo Drive Setup Procedures

The following procedure is only to be used to make the servo drive operational.

1. Apply three phase power to drive <u>without</u> encoder connected. It is fine if the motor power to motor is connected.
2. Connect a serial cable with a null modem adapter attachment to the servo drive in the RS232 communication port
3. Start Maxforce software (also referred to as Launchpad.exe)
4. Drive should be detected upon software launch and will ask to run initial setup. Say yes.
5. To select the model number, use model number G415-800APTC and <u>not</u> the actual model number. This is a linear actuator model with the same setup parameters. Click on “use new model”.
6. Follow on screen instructions for booting and re-booting.
7. Turn power back on for the drive. <u>If the drive enable is active prior to startup, the drive will not start. If this happens, the drive will need to be reset to its factory default settings.</u>
8. Connect 24V to the drive enable pin (J2A pin 1) and GND to RET on J2A.

9. The drive/motor will most likely fault. Adjust the Velocity Error Fault Limit which is found in the Control panel tab > Diagnostic Window > Select.
10. The Static to Dynamic Threshold should be somewhere near the Velocity Error Fault Limit. Refer to the manual for exact specifications. (file: CDS7323-A EMA Commission Software Manual)
11. The drive might still spin on its own without any move parameters. Refer to the DS2110 Commutation Offset Procedure below if this occurs.

To have a set up similar to the Phase II Senior Design Team, perform the following steps.

1. Go to the Motion tab and Select Move 1.
2. Check Move Position 0-100% in the Analog input function.
3. The team used Command Source Select with -10 to +10 Vdc. This is misleading as the voltage response does not exceed 4.7V.
4. The Command/Feedback Function was set to -10V Retract – 10V Extend.
5. Command Deadband was set to 0.
6. Move position was set to 4, speed set to 1000, and accel/decal time set to 1 second.
7. I/O setup must be pins 1, 2 and whatever input is assigned to move start set to active high (24V). Without move start, move 1 will run once and stop.
8. Refer to the analog I/O portion of the main drive manual (CDS7324) for pin to analog I/O. Keep in mind anything past 4.7V had no effect on the drive used by the Phase II team.
9. Use any analog input control to control the motors with respect to position. (Potentiometers, digital potentiometers, give PWM a try)

7.3 DS2110 Commutation Offset Procedure[31]

1. Turn on AC power to drive
2. Make sure actuator is in a position to move in both directions
3. Go to Control Panel tab
4. Select Diagnostics Window
5. Select Access Control
6. Enter password: 3287
7. Select Submit
8. Verify Level is now at 8
9. Close Access Control Window
10. Select System Commands
11. Disable Model
12. Verify status LED 2 stops blinking on the drive
13. Select Commutation
14. Select Encoder Feedbackf or the Commutation Type
15. Turn on 24Vdc hardware enable
16. Select Auto-Run Commutation Offset Routine
17. Once completed, select OK
18. Select Position Mode
19. Close Encoder parameters window

20. Turn off 24Vdc hardware enable
21. In System Commands, Enable Model
22. Verify status LED 2 is blinking again on the drive
23. Select Save Parameters
24. Close System commands and Diagnostics windows

7.4 Servo Motor and Drive System Components

Component Name	Component Part Number	Part Included? Y/N
Rotational Actuator	G405-1034A	Y
DS2110 Analog Drive	G362-020-70-C-902G	Y
3 Phase Power Cable	CA28936-002-005	Y
Encoder Cable	CA65132-003-005	Y
12Ω Power Resistor	NA	Y
RS232 Null Modem Cable	NA	N
9 Pin Male Connector	NA	N
Wire Crimps	NA	N

Table 7. Servo Motor and Drive System Components

7.5 Motor Programming and Communications

The following Arduino sketch code is what was used to communicate with all the system motor drives. The top section of define statements sets up the Arduino UNO microcontroller pins with desired pin names to make the program more legible. The setup function establishes the baud communication rate and establishes the modes of pins as either inputs or outputs. After setup, the mapping variables are initialized. Finally, the loop function is run continuously throughout simulator operation. It reads x-y coordinates from the dual joystick potentiometers and relates those to a value established by the mapping function.

Based on the joystick inputs, the program runs through a number of logical if-then-else structures to determine what type of movement should occur and how highly scaled this value should be. This is written to the pins and communicated to the devices by simple analog communications. The program is delayed to allow for processing, feedback signals and physical events to occur, and then repeats.

```
//Simulator Control Program
#define speedPin1  9
#define speedPin2  10
#define dirPinRev1  2
#define dirPinFwd1 3
#define dirPinRev2  4
#define dirPinFwd2 5
#define JoyX  A0
#define JoyY  A1
#define inductionPot1 A2
#define inductionPot2 A3

void setup() {
 // put your setup code here, to run once
 Serial.begin(9600);
```

```
 pinMode(dirPinFwd1, OUTPUT);
 pinMode(dirPinRev1, OUTPUT);
 pinMode(dirPinFwd2, OUTPUT);
 pinMode(dirPinRev2, OUTPUT);
 pinMode(speedPin1, OUTPUT);
 pinMode(speedPin2, OUTPUT);
 delay(50);
}

// variables for mapping
int speedRoll_1 = 0;
int speedRoll_2 = 0;
int readX = 0;
int readY = 0;
int testWrite = 0;
int MAX_PWM_OUT = 200;
int MIN_PWM_OUT = 25;

void loop() {
// put your main code here, to run repeatedly:

// read and map the values from a joystick for testing purposes
readX = analogRead(JoyX);
readY = analogRead(JoyY);

Serial.print(speedRoll_1);
Serial.print(";");
Serial.println(speedRoll_2);

digitalWrite(dirPinFwd1, HIGH);
digitalWrite(dirPinRev1, LOW);
digitalWrite(dirPinFwd2, LOW);
digitalWrite(dirPinRev2, HIGH);

speedRoll_1 = map(readX, 0, 1023, MIN_PWM_OUT, MAX_PWM_OUT);
speedRoll_2 = map(readX, 0, 1023, MAX_PWM_OUT, MIN_PWM_OUT);

analogWrite(speedPin1, speedRoll_1);
analogWrite(speedPin2, speedRoll_2);

delay(1);
}
```

8 Scissor Arm Design

8.1 Scissor Arm Overview

The three scissor arms cushion the simulator as it mimics a plane losing sudden altitude due to descent or turbulence. Each arm has two separate members that are connected to each other by means of a pin joint. The lower member connects to the shock system spring and the simulator base. The upper member connects to the central universal joint structure below the platform. The springs connected to the lower arms are what creates the “cushioning effect” for the simulator when it moves

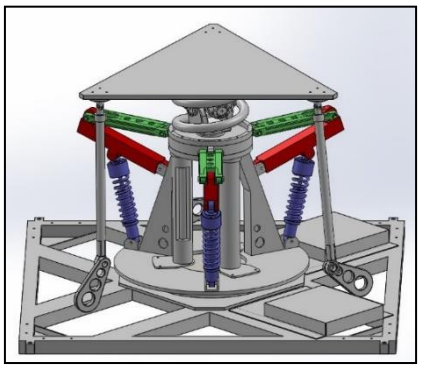


Figure 29. Scissor arm system in Solidworks model of simulator

in the z-direction. The scissor arm system is shown in Figure 29. The upper scissor arms are shown in green, the lower scissor arms are shown in red and the shock absorbing springs are shown in blue.

8.2 Purpose of Redesign

The previously manufactured lower scissor arm was incorrectly designed and over-engineered. The existing upper scissor arms from the Gyro IPT™ consist of welded aluminum plates and internal pin structures. The existing lower scissor arm is made of 1.5in steel tubing with welded on pin tabs. The built-in safety factor of this component is nearly seven times the failing criteria of the upper scissor arm. The arm adds significant weight to the lifting of the central spring unit, with each arm weighing approximately 10lbs. The scissor arm also collides with the motor while in rest position and does not allow for the unit to rest on its base when turned off, as shown in Figure 30 in the red box. The redesign uses less material, eliminates interference between components and was designed based on the failing criteria of the upper scissor arm to avoid over-engineering.

Figure 30. Old lower scissor arm installed on system. Notice gap between upper ring and lower ring on base indicated by the red box.

8.3 Scissor Arm Theory and Free Body Analysis

To perform the finite element analysis on either scissor arm, it is necessary to understand how the loadings are applied to the arms, the different failure scenarios that could potentially break the arms and the different assumptions that need to be made to design a stable system. A free body analysis of the entire scissor arm system is shown in Figure 31. For the analysis, the upper scissor arm is the point of interest. There are two forces acting on this component and therefore it can be simplified into a two force member, shown on the bottom of Figure 31.

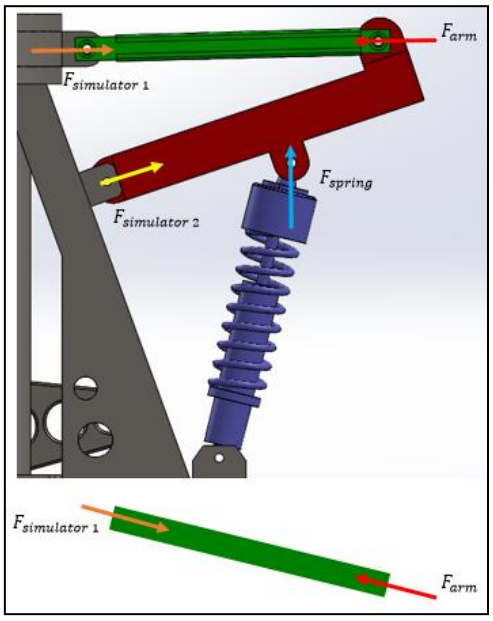


Figure 31. Free body diagram for upper scissor arm analysis

From the free body diagram, it can be seen that the upper scissor arm is in compression. The forces acting on the arm are a portion of the simulator weight and the force of the lower arm pushing back on the upper arm at the pin joint. To further simplify the loading case, one pin was assumed to be fixed while the other provides a compressive load against the bearing hole in a direction purely along the arm, Figure 32. Realizing that the load is being concentrated on the bearing holes rather than the end of the arm itself, a high concentration of stress inside of the bearing holes was

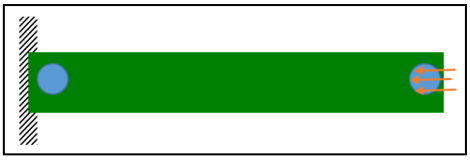


Figure 32. Upper scissor arm simplified into a two-force member

expected. Therefore, the arm will either fail in buckling or at the bearing holes due to yielding[5].

8.4 Upper Scissor Arm: Finite Element Analysis

8.4.1 Model Selection

Prior to analysis in ABAQUS, it was crucial to ensure the upper scissor arm drawn in Solidworks included every detail of the member including the internal support pins. Through observation of the model and understanding of the finite element analysis (FEA) software it was determined that the model could be simplified. Initially, there were three points of interest to consider in the upper scissor arm analysis[8]:

1. The welds connecting the plate to the member
2. The pins at each end of the member
3. The plates on each side of the member

The total member is shown in Figure 33 with just the plates shown in Figure 34.

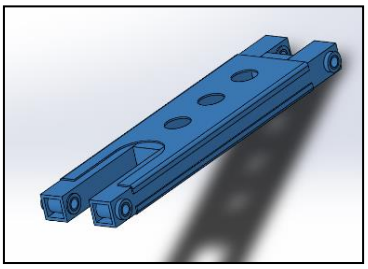


Figure 33. Entire upper scissor arm modeled in Solidworks

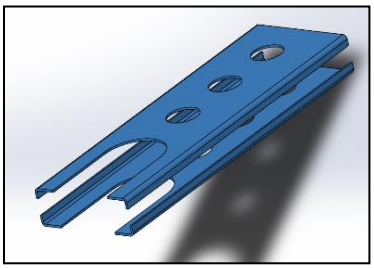


Figure 34. Side plates of upper scissor arm modeled in Solidworks

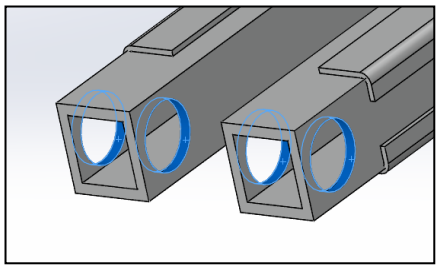


Figure 35. Bearing stress area shown in Solidworks model

Of the three points of interest, it was determined that the welds would be the most difficult to model. To model these points the material properties needed to be altered around specific sections, which complicated analysis. The welds are not interfering with the location that the load is being applied or the area that the member is fixed therefore the decision was made not to model the welds in SolidWorks and ABAQUS. Although the decision affects the analysis results, there will not be a drastic difference.

The second point of interest were the pins that sit in the bearings which allow the arms to pivot about a point. Too assess if the pins were necessary in the FEA model, the load distribution between the pins and member was explored. The load is transferred through the pins only at the location where the pin is in contact with the member. Assuming that the majority of the load is then transferred onto the bearing hole surfaces rather than the pin itself, the pins can be safely removed without drastically impacting the results of the FEA analysis. The area where the force will be distributed is shown in Figure 35.

The third point of interest in determining the model used for FEA are the plates on either side of the upper scissor arm. These plates are welded to the member on either side and provide extra support to the member during loading. Since it is unknown how both loading cases will be impacted by keeping or removing these plates, two models were created; one with the plates as part of the solid model and one without.

8.4.2 Analysis Criteria

The program used for analysis was ABAQUS CAE[1]. When compared to ANSYS, another option for FEA, ABAQUS provides better meshing capabilities and a user-friendly interface more intuitive to a new user than ANSYS[8]. In order to achieve accurate and usable results, it was imperative that appropriate inputs were provided including material properties, boundary conditions, loading scenarios and meshing and mesh convergence. These inputs are outlined in the subsections below.

Material Properties

The material used to manufacture the upper scissor arm was 6061 Aluminum. The relevant material properties of this material include a Young's Modulus of 69x10^9 $6969x10^9 \frac{N}{m^2}$ and Poisson's Ratio of .33.[29] These properties were assigned to all parts of the upper scissor arm as it was all manufactured using the same material.

Boundary Conditions

The boundary conditions used in the analysis were difficult to determine, as the decision was made to exclude the pins from the analysis. This created a challenge regarding fixing one end of the member in order to constrain it from rotating or moving about the x, y, and z-directions. To determine the quality of the chosen boundary conditions prior to analysis, three working engineers were consulted as resources[8]. The most accurate way of performing the analysis was discovered to be through fixing one set of bearing holes. The holes were fixed on one half of the surface, as only that half would be in contact with the pin during compression. This is depicted in Figure 36.

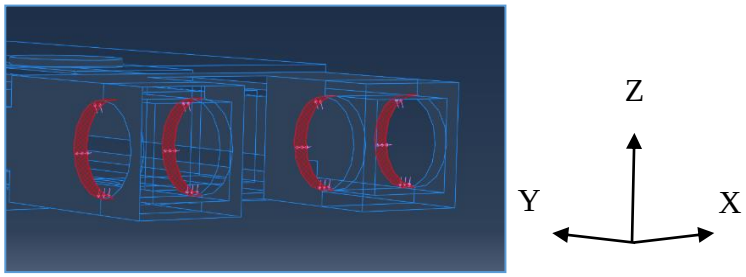


Figure 36. Bearing stress contact surface in Abaqus

Loadings

The majority of an effective and accurate FEA depended on understanding the way that the component was loaded, therefore determining the way the member deformed at failure. The member is subjected to a compressive force that acts on the bearings, which indicates a bearing stress scenario. This loading can be summed up using the bearing stress equation:

$$\sigma_b = \frac{P}{A_b} \qquad \textit{(Equation 17)}$$

Where A_b is the surface area of the hole, or in this case the diameter of the hole being multiplied by the thickness of the hole and P is the load magnitude. This loading is shown in Figure 37.

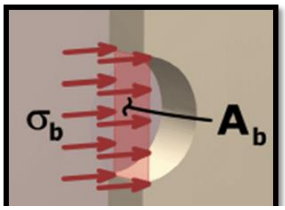


Figure 37. Bearing stress loading

Since the pin was omitted from the analysis, the bearing stress was assumed to distribute uniformly over the bearing hole. As a result, during the FEA the load was applied to half of the bearing hole as shown in Figure 37. This caused deformation only to occur on the z-axis and created an accurate output stress concentration in the bearing holes.

Meshing and Mesh Convergence[21]

ABAQUS requires the user to mesh the part based on the number of seeds along each edge. Meshing the member was simplified due to the symmetry in the part, therefore ordinary meshes could be used for the FEA. The mesh convergence was critical in this analysis because depending on the mesh density size an incorrect maximum stress can be evaluated. This phenomenon is shown in Figure 38 and Figure 39. To test mesh density, a mesh convergence test was performed to compare the amount of elements in the mesh to the maximum stress output. The mesh density was determined to be the location of where this plot leveled out. The number of elements required for this analysis was evaluated to be 9038. The mesh convergence graph is shown in Figure 40.

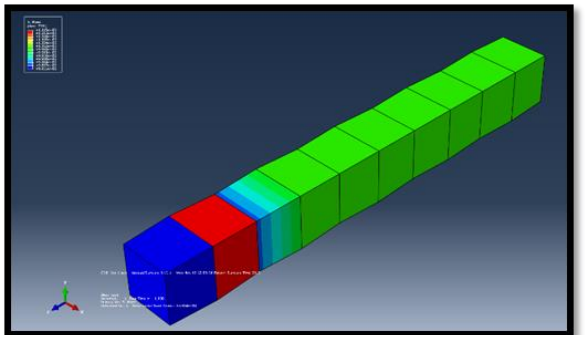


Figure 38. Stress analysis of beam with low mesh density in ABAQUS

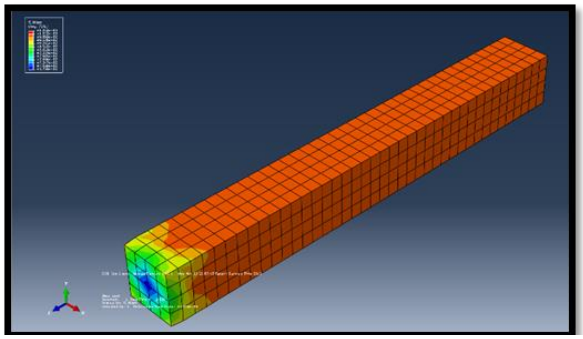


Figure 39. Stress analysis of beam with high mesh density in ABAQUS

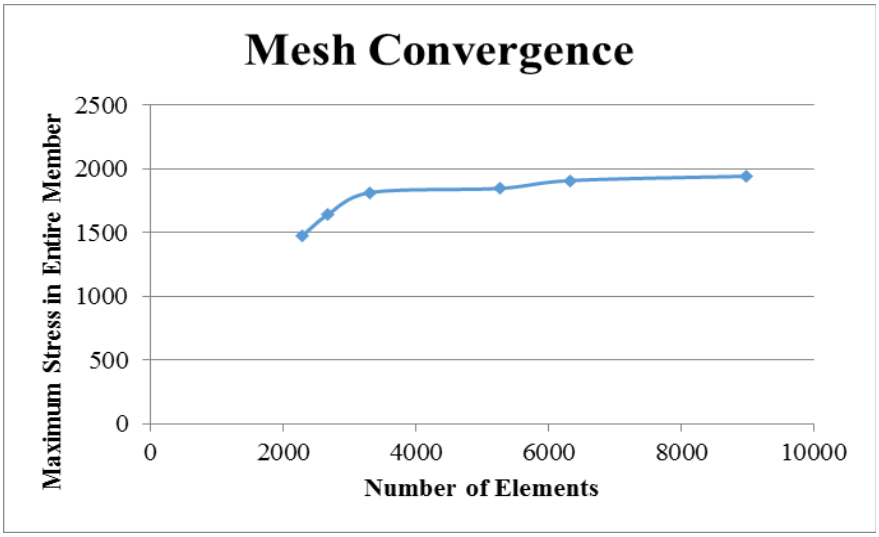


Figure 40. Mesh convergence graph of ABAQUS iterations

8.4.3 Analysis Results

The FEA was completed using the aforementioned criteria. Using 9038 as the number of elements in the mesh density resulted in a stress distribution shown in Figure 41. It can be clearly seen that the member will fail at the bearing holes before any other member deformation. This failure occurs at approximately 2400 N, or 540 lbs, for each upper scissor arm. The maximum weight of the simulator with a pilot is approximately 1240 lbs, therefore each arm will have an additional 126 lbs beyond the maximum weight that they can hold before failure occurs.

The results of the FEA allow the design, analysis, and manufacturing process of the lower scissor arms to match the characteristics of the upper scissor arm. This will help to prevent over-engineered components, thus saving money.

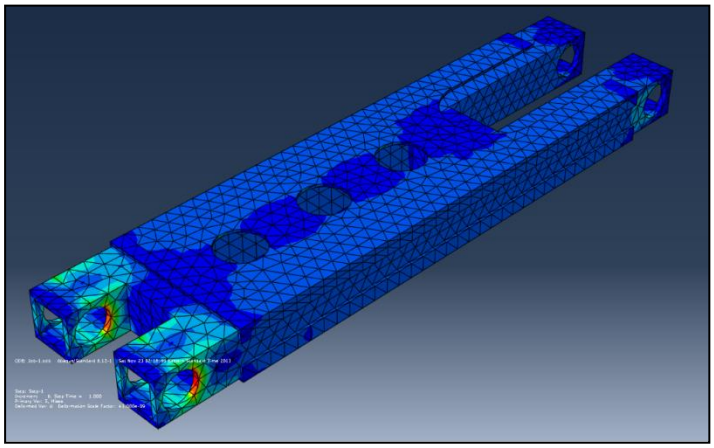


Figure 41. Stress distribution on upper scissor arm

8.5 Lower Scissor Arm Design

8.5.1 Parametric Model Development

The parametric model of the simulator base was used to design the new lower scissor arm with the criteria found through the analysis of the upper scissor arm. An important design aspect of the arm includes the lower tab coinciding with the shock absorbing spring. Each spring bottoms out at 9.24” and is fully extended at 13.01” so each tab must allow for the spring to fit comfortably within this range. The arm must also allow for clearance above the existing induction motors, as well as the proposed servo motors. In order to accommodate the length requirements the team decided to utilize the 3D printer available on campus and create an adjustable arm, seen in Figure 42. This arm contained a threaded stud in the center allowing for length adjustments which could tweak the overall size of the arm without wasting the raw material purchased by the sponsor. The final lower scissor arm design is shown in Figure 43.

Figure 42: 3D Printed Adjustable Lower Scissor Arm

8.5.2 Finite Element Analysis: Model Selection and Criteria

After the lower scissor arm was designed, it was necessary to perform finite element analysis on it in order to confirm that it can support the weight of the simulator as well as other combining forces. To begin the analysis, a free body diagram was used to visualize how the forces would act upon the simulator. The same free body diagram shown in Figure 31 can be used, only with the compressive forces in the two-force member reversed to represent the tensile forces on the scissor arm.

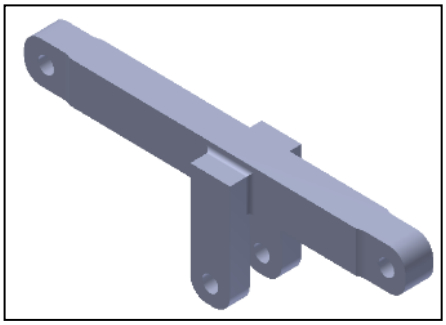


Figure 43: SolidWorks Model of Lower Scissor Arm

After reviewing the free body diagram, we chose to model the arm under tensile stress at each end as we found that the arm would fail first due to tensile or compressive stresses at the bearing holes rather than to bending near the spring force. The forces being applied are very similar to the forces that acted upon the lower scissor arm earlier in this report. Therefore, the analysis would be carried out in a very similar manner as well. Keeping in mind that the arm was analyzed in tensile stress, it was found that we could also perform a real tensile test on a quarter scaled scissor arm in order to verify that we were performing our analysis correctly.

The material chosen to design the scissor arms with was Aluminum 6061-T6. This material is similar to the one that the lower scissor arms are made of, and we assumed that it would be strong enough to support all of the forces acting upon the newly designed scissor arm. If, after the analysis, it were not strong enough, a new material would be chosen in accordance with the analysis. The properties mainly looked at in our design were the tensile yield and ultimate strength, Young's Modulus, and the Poisson ratio. These values can be seen in Table shown below:

Property	Value
Young's Modulus	68.9 GPa
Poisson's Ratio	0.33
Tensile Yield Strength	276 MPa
Ultimate Tensile Strength	310 Mpa

Table 8: Properties for Lower Scissor Arm Analysis

8.5.3 Analysis Results

The newly designed scissor arm was analyzed in tension using ANSYS software to perform the analysis using the loadings as shown in Figure 44.

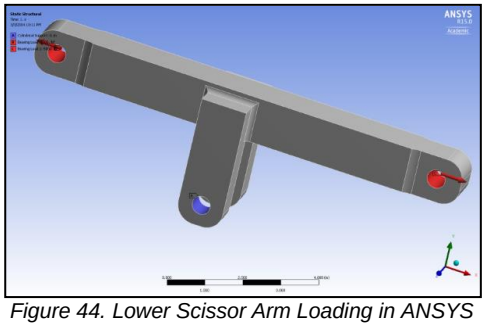


Figure 44. Lower Scissor Arm Loading in ANSYS

As can be seen, the two end bearing holes are being pulled in tension while the bottom two bearing holes are used to fix the arm in place. It can be noted that a mesh convergence analysis was completed for this arm, just as one was completed for the lower scissor arm earlier in the report. A high mesh density was chosen after reviewing the mesh convergence analysis. After the simulation of the loading was complete, we were left with an arm that would look as follows in Figure45.

The highest concentration of stress is at the sides of the bearing holes, which was expected. There is also little to no deformation in any part of the arm besides at the ends. The final results of the analysis report that the arm fails at the bearing holes at about 5,000 lbs in shear as a tensile load is applied. The shearing would occur at the highest stress concentration points, which are at the sides of the bearing holes rather than at the top or the holes

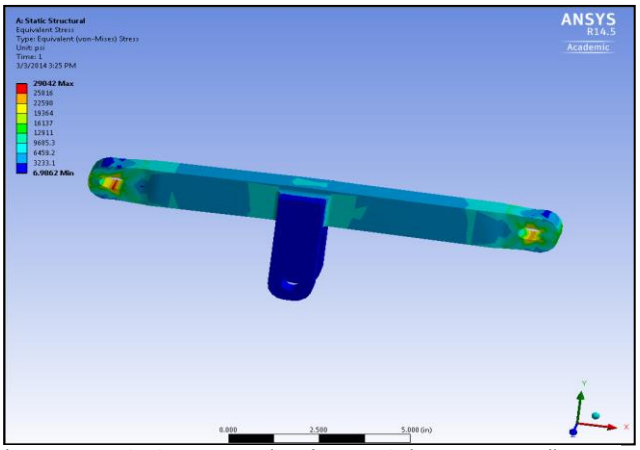


Figure 45: ANSYS FEA Results of Lower Scissor Arm Loading

8.5.4 Tensile Test Validation

To verify the accuracy of the analysis, the team manufactured a quarter-scaled scissor arm, analyzed it in ANSYS, and performed a physical tensile test on it to see if the results matched. The quarter-scale arm was inexpensive, and would served a huge purpose in this multi-year capstone project, as it verified that the finite element analyses. This arm is shown in Figure 46.

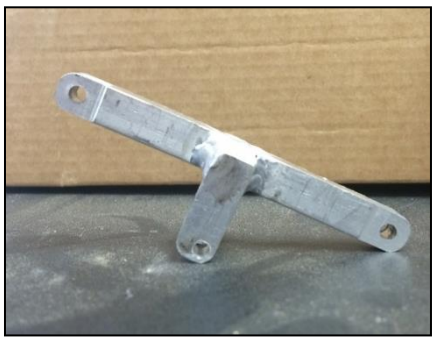


Figure 46: Quarter-Scale Model of Lower Scissor Arm

After the arm was modeled into ANSYS, identical results were found for the full-scale scissor arm, except the failure loadings were scaled differently as the arm was much smaller. We found after FEA, the quarter-scale arm would fail/yield around a 600 lb. maximum applied force, or 17,000 psi. The arm was expected not to fail at the very tip of the bearing hole either, yet at the sides of the bearing hole; most likely just one side, and not both at the same time due to imperfect manufacturing.

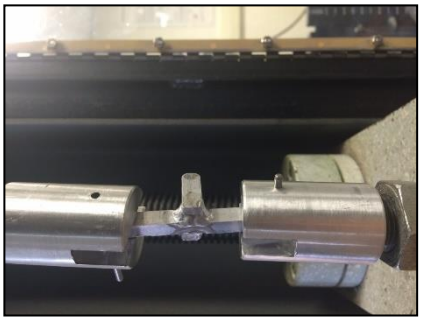


Figure 47: Scale Model Arm Fixture

The quarter-scale arm tensile test required that the arm be fixed slightly differently than what was modeled in the analysis, but the results were expected to be accurate. The fixture that was used to hold and load the arm is shown in Figure 47. Instead of having the bottom bearing holes fixed by the "springs", one end of the arm was fixed while loadings were applied to the other.

Once the physical tensile test was performed, the results were acquired. From the tensile test, it was found that the quarter-scale arm failed in shear at a 45 degree angle to the hole, as can be seen in Figure 48. This showed that the highest concentration of stress was in fact not at the tip of the bearing holes, but rather at a 45 degree angle to the side or the top of the holes. This deviation from the analysis could have been due to the fact that our loading was slightly different than what was in the FEA, yet it is also found that for aluminum components stressed in tension this was typical.

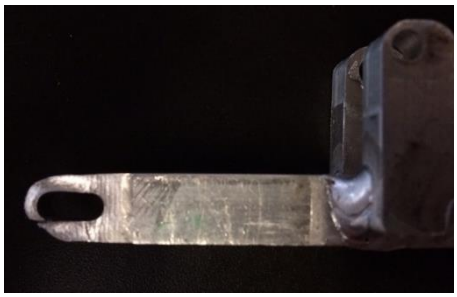


Figure 48: Failure of Scale Model

The final results for the physical tensile test were that the quarter-scale arm failed in shear at a 45 degree angle at a max loading of about 600 lb. The stress-strain curve can also be seen in Figure 49, which also shows that yielding begins at approximately 17,000-18,000 psi. These values accurately match the expected results from our FEA, which verified the scissor arm FEA.

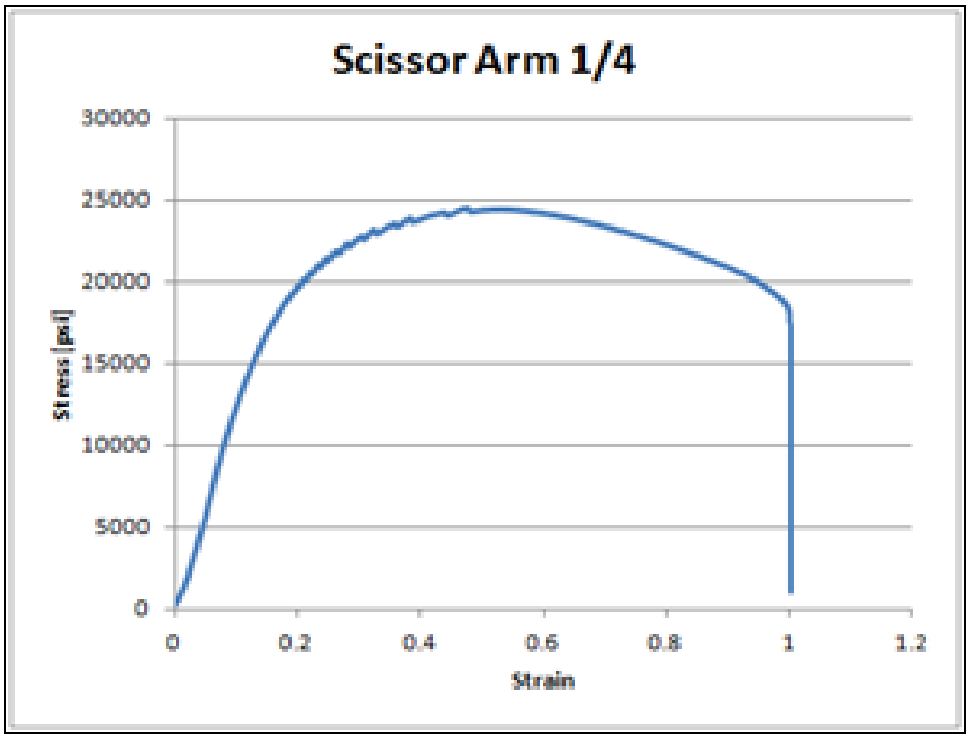


Figure 49: Stress-Strain Curve of Scale Model Failure

8.5.5 Finite Element Analysis Results

From the results of our FEA, the upper scissor arm, manufactured from aluminum 6061-T6 material, fails at the bearing holes around an applied force of 5,000 lb. Since the maximum weight the simulator ever imposes on the bearing holes is approximately 1,250lbs, the scissor arms are strong enough to support forces that act upon it. These results were verified by performing a physical tensile test on a quarter-scale arm and found that the results obtained matched a similar FEA performed on a quarter-scale arm.

9 Motor Mount Design

9.1 Motor Mount Overview

The existing motor mounts are shown in Figure 50 and Figure 51. The existing motor mounts are sufficient for securing the induction motors, but they are inadequate for securing the servo motor and gearboxes that will eventual replace all the induction motors. The mounts are critical components as they hold the motors in place and support the entire simulator weight while in operation.

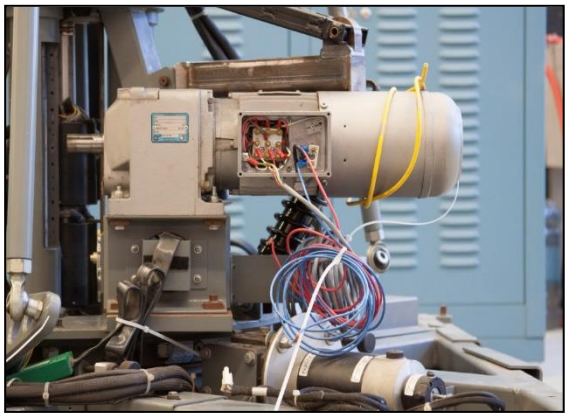


Figure 50: Existing Motor Mount with Induction Motor

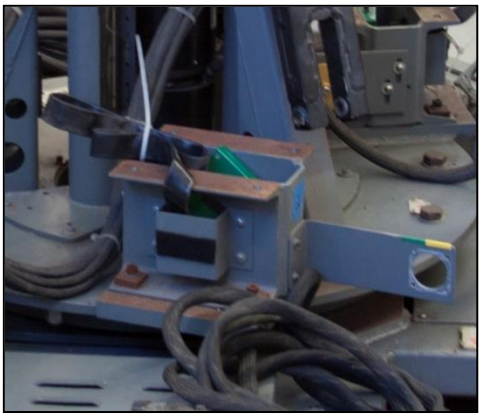


Figure 51: Existing Motor Mount without Motor

9.2 Parametric Model Development

The parametric model of the simulator base was used again to design the motor mounts. The concept for the mounts was to have a single base plate that bolts onto the existing motor mounts, a face plate that would bolt onto the gearbox face and weld into the base plate, and two side plates to add support to the whole bracket. The motor mount was then pinned and welded in the machine shop.

When designing the parametric model, there were critical dimensions to consider. First, the gearbox face contains a hole pattern situated on a 4.252” diameter bolt circle, that needed to be replicated on the new face plate. These holes also needed to have an adequate countersink in order to avoid interference with the cam arm. Second, it was critical in the design of the new motor mount that the cams sit in an identical position on the servo setups as they did on the induction setup. This was verified by matching critical dimensions from the original motor setup and translating them into the new motor mount bracket design. Finally, the parametric model needed to be simple enough to manufacture with minimal waste, however it also needed to be strong enough to withstand the forces of the simulator motion. The final parametric model of the motor mount is shown in Figure 52. To understand the forces acting upon the motor mounts, a free body analysis was performed.

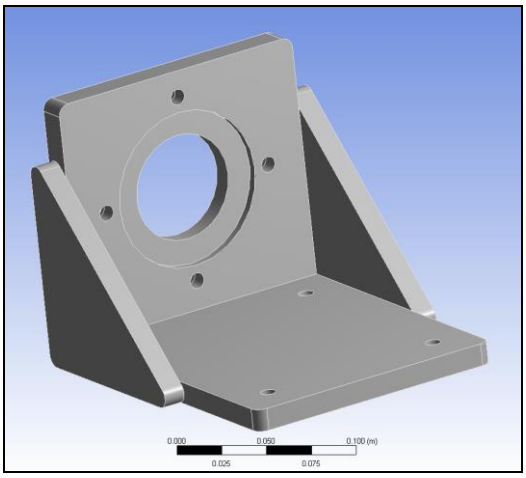


Figure 52: SolidWorks Model of Motor Mount

9.3 Finite Element Analysis

9.3.1 Model Selection

Once a free body analysis was carried out on the motor mounts, two main forces needed to be analyzed: the simulator weight force and the motor spin torque. These forces have been illustrated in Figure 53. What this meant was that we would need to simulate in ANSYS the main face of the motor mount having both these types of forces acting on the bearing holes at the same time.

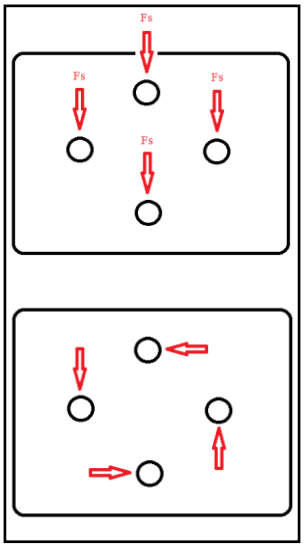


Figure 53: Motor Mount Forces

To simulate the forces, bearing loads were placed at each of the holes splitting the weight of the simulator (Figure 54a) and added four other bearing loads separately to depict the torque acting upon the bearing holes (Figure 54b). To fix the motor mount, the bearing holes on the bottom side of the plate were fixed, as these were to be attached to the simulator base (Figure 54c). These do not experience as much force as the main face of the motor mounts so their analysis was unnecessary.

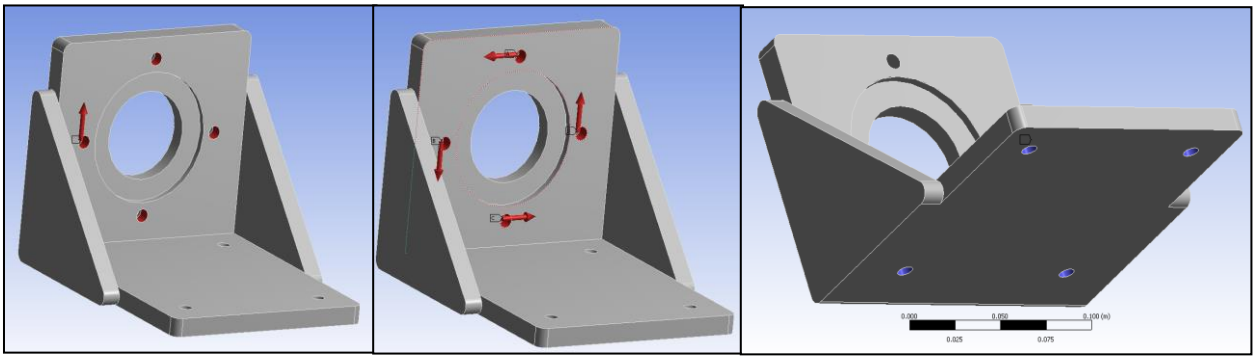


Figure 54: Motor Mount Loading: (a) Splitting bearing load at each hole (b) adding torque to bearing holes (c) fixing motor mount

One thing to note about this FEA is that the simulator weight may not always exclusively act in the downward direction, as the motors move and cause the force to be applied in slightly different ways. Therefore, eight different simulations were performed. In these simulations, the direction of the weight acted in the north, northeast, east, southeast, south, southwest, west, and northwest directions. From these simulations an accurate reading of how much stress the motor mounts should handle.

9.3.2 Analysis Criteria

The material chosen for the motor mounts was either Aluminum 6061-T6 or Steel 8620. The steel material was chosen as it was much stronger in case the Aluminum was not strong enough to support the forces applied to the motor mounts. For the analysis, only one of the directional cases for the steel was performed as it was significantly stronger. The purpose of this analysis was to ascertain approximately how much more stress steel can handle.

9.3.3 Analysis Results

After performing the FEA, results for the eight different loading scenarios using Aluminum 6061-T6, as well as one loading scenario (the direction of the simulator weight was chosen based on the most-likely-to-fail case found from the aluminum trials) for steel 8620 were aquired. These analyses were performed assuming our maximum weight from the simulator imposed on the motor mounts would be about 1200lbs, and the torque force would be about 6261.95N at each hole (this was taken from the max torque that the motors output).

The results obtained from each scenario showed that if aluminum were used, the motor mounts would be able to withstand the stress, but with a factor of safety range of 2.365 to 2.905. The steel, on the other hand, withstood these stresses with a factor of safety of 5.334, which was significantly higher. You can as well see in Figure 55 the stress concentrations being mainly on the bearing holes, with little to none elsewhere.

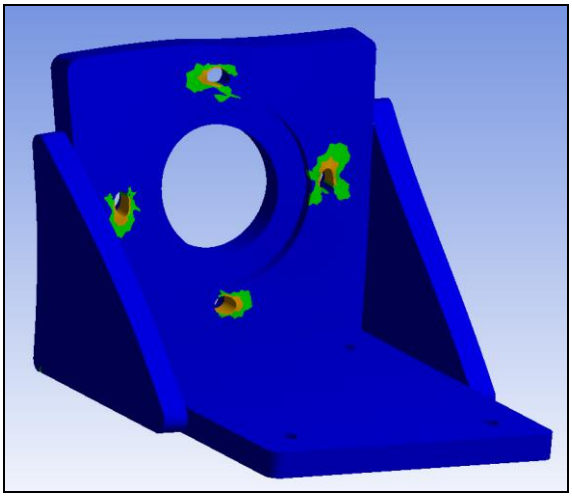


Figure 55: ANSYS Results for Motor Mount

9.3.4 Motor Mount Analysis Conclusions

From the obtained results, assuming manufacturing errors, steel 8620 was used for the motor mounts. This material surely accommodates for any errors in manufacturing, ensuring the motor mounts are be safe. Using Aluminum 6061-T6 would be satisfactory, but is ultimately too much of a risk to use this material as the team is custom machining the parts.

10 Conclusion

10.1 Summary of Project Accomplishments

Connecticut Corsair's overall goal for the University of Connecticut joint discipline senior design team was to restore a damaged Gyro IPT flight simulator with obsolete components to working condition. The 2013-2014 senior design team picked up where the last team had stopped and formed the overarching goal of restoring motion to the simulator in three axes, pitch, roll, and heave. The overarching goal was broken down into mechanical and electrical engineering distinct deliverables.

The first mechanical deliverable was to establish the torque requirement for the simulator in order to facilitate the selection of a new gearbox and motor combination. The second mechanical deliverable was to redesign the lower scissor arm, which eliminated interference with the simulator as well as the existing over engineered design. The third mechanical deliverable was to create motor mount attachments in order to secure the newly selected gearboxes to the simulator base. The final mechanical deliverable was a conversion of the entire base into a working SolidWorks assembly.

The first deliverable was broken down into several smaller goals. The first goal was to determine the torque, angular velocity, and angular acceleration requirements of the new motor. To determine each of these requirements a free body diagram was drawn of the entire simulator base. The free body diagram indicated that there were two distinct motion profiles of the simulator; a vertical lift and a pitch/roll motion profile. Each of the distinct motion profiles with their corresponding free body diagrams can be seen in Figure 15 and Figure 19. The distinction between the motion profiles can be seen in the respective equation of motion, Equation 7 for the vertical lift motion profile and Equation 10 for the pitch/roll motion profile. Equation 7 defines torque on the cam with only one term because during vertical lift, all cams must rotate symmetrically and there is only one force from the pushrod acting on the cam. For the pitch/roll motion profile, however, there are two terms within the equation of motion which are defined by the force from the pushrod acting on the cam and the restoring force of the spring acting on the platform. In the pitch/roll profile the simulator platform rotates about the universal joint and compresses and extends the spring surrounding the universal joint. This spring acts as a restoring force, constantly trying to return the platform to level. After defining the motion profiles with the equations of motion, the spring constant as well as the motion relationships of the simulator had to be defined.

Each of the equations of motion contained multiple dependent angle relationships which can be seen in Figure 15 and Figure 19. Each of the dependent angle relationships could not be defined

by linear relationships with each other, forcing the development of two CAD models. The first CAD model was of the vertical motion profile and modeled the relationship between the cam, pushrod, and simulator platform. The second CAD model was of the pitch/roll motion profile and modeled the relationship between the cam, pushrod, and simulator platform. Each of the models were used to produce the graphs in Figure 17 and Figure 21, which depict the nonlinear relation between the rotation of the cam and all other dependent angles. MATLAB was utilized to curve fit the data and produce functions relating the angle of rotation of the cam to the other dependent angles.

The next step that was undertaken was to experimentally derive the spring constant of the restoring spring. To derive the spring constant, an experiment was conducted in which varying weights were hung from the edge of the simulator, Figure 13, and the vertical displacement of the platform of the simulator was measured. The intention behind hanging weights from the simulator was to replace the force of the pushrod with a measureable force, a hung weight. Then the displacement verses force data was graphed and produced Figure 14. The graph revealed a linear relationship between force and displacement allowing the spring equation $F = kx$ to be used and the spring constant to be defined as 189000 N/m.

Once the motion profiles, free body diagrams, equations of motion, CAD models, dependent angle relationships, and spring constant were determined a MATLAB program was written to combine all of the data. The MATLAB program created two torque curves, one for the vertical lift case, Figure 18, and one for the pitch/roll case, Figure 22. Analysis of the two different plots revealed the maximum torque requirement to be 3500 in-lbs. Also the data was utilized to calculate the maximum angular velocity of the motor to be 85 RPM.

The motor requirements were validated in three separate ways. The first validation was done by a Moog Motor Corporation product application manger of their simulation division. The product application manager had designed a cam driven simulator in the past that was comparable in weight and size. In the design of the simulator Moog had utilized servo motors that were rated at 3500 in-lbs. Therefore, since the two simulators are comparable in size and the derived torque requirements match, the torque requirements are validated. The second validation was done by the COO of Environmental Tectonic Corporation. The 3500 in-lbs. torque requirement was presented to the COO who verified that the torque requirement was reasonable. The final validation for the torque requirement was done by comparing the torque requirement of the old motor to that of the derived torque requirement. The old torque output was 1212 in-lbs. which is less than half of the new torque output. Therefore, the new torque requirement which is more than double the previous torque requirement insures that the simulator will function.

After validation, the motor requirements were presented to Moog Motor Corporation who then presented the G-5-M8 motor as the correct motor for the application. Then Neugart was presented the requirement of needing a 20:1 gearbox with the loading scenarios of a cam driven simulator. Neugart presented the WPLPE120-20 as the correct gearbox for the application.

The second deliverable was to redesign the lower scissor arm so that it was not overbuilt and did not collide with the base. Redesign of the lower scissor arm began by drawing the free body

diagram of the entire scissor arm assembly. The free body diagram allowed the simplification to be noticed that the failure loading of the upper arm should match the failure loading of the lower arm. The reason behind the simplification is if the lower arm has a higher failure loading point than the upper arm, then the lower scissor is overdesigned. In reverse, if the lower scissor arm has a lower failure loading point then it becomes the weak link in the system. In order to determine the failure loading point of the upper arm the arm was reconstructed in CAD, Figure 33. Then FEA analysis of the upper scissor arm model was done in ABAQUS.

The first step in the FEA analysis was to determine the loading scenario that best represented the loading scenario in real life. After analyzing the free body diagram again, the best representative loading scenario was to load the upper scissor arms at the bolt holes in pure compression along the longest axis of the part. Once the loading scenario was chosen then a mesh convergence was performed to determine the optimal mesh density for the part, which was 9038. Finally the ABAQUS simulation determined that the failure for the upper scissor arm is 540 lbs.

After determining the failure load for the upper scissor arm, the loading of the lower scissor arm was analyzed revealing that the lower scissor arm fails in tension at the bolt holes. The maximum tension that the lower scissor arm must withstand is 540 lbs. because the upper scissor arm will fail after this point. A design for the lower scissor arm was created in SolidWorks and then 3-D printed out of ABS plastic to check for collision interferences on the simulator base.

In order to validate all of the ANSYS modeling a quarter scale model of the lower scissor arm was machined and an FEA analysis was performed. The FEA analysis consisted of loading the quarter scale model in tension along its longest axis until failure, exactly the same as the full size part. Failure of the quarter scale model in ANSYS was determined to be between 550 lbs. and 600 lbs. To validate the FEA model the machined quarter scale model was placed in a tension test machine and was pulled apart until it failed at 600 lbs. Both the real world model and the ANSYS model failed at the same load validating the FEA model. After validation, the full scale lower scissor arm design was validated in ANSYS with a failure loading of 5,000 lbs.

The failure loading of the lower scissor arm is considerably higher than the failure loading of the upper scissor arm which allows it to be a success and overdesigned. Though the piece is overdesigned, the design optimized the manufacturing and material resources that were available. The team was limited to what could be manufactured in the machine shop and also designed the part with ease of manufacturing in mind. Therefore, the part was overdesigned but succeeds in the overall goal of producing a lower scissor arm that works.

The third deliverable of the project was to design a new motor mount for the gearbox and motor combination. In order to design the new motor mount, the full assembly of the base in SolidWorks was utilized. Both the model of the gearbox and the motor were imported into the SolidWorks assembly which allowed the visual placement of the new motor and gearbox within the model. Then certain key dimensions such as the distance from the top of the old motor mount to center of the gearbox spindle and distance from face of the old motor mount to face of the gearbox spindle were constrained. These dimensions were important because they placed the spindle of the gearbox in the same exact place as the spindle from the old gearbox, allowing the

simulator to function correctly. Once the gearbox and motor were constrained a new motor mount was designed utilizing the attachment points of the old motor mounts. Then an FEA simulation in ANSYS was performed that applied the maximum torque and maximum force onto the motor mount. The lowest safety factor on the motor mount was 2.4 signaling that the design is sufficient.

The final deliverable for the mechanical engineering team was to convert the entire base of the simulator into a SolidWorks assembly. The entire base was measured and converted into a SolidWorks model which can be seen in Figure 10. Overall the model aided in the design of the previous deliverables.

The electrical engineering team had three distinct deliverables to meet the goals of the Corsair Design Project. To summarize, the three deliverables were: the repair and installation of the induction and servo motors, providing user controllability to the simulator, and a study and preparation of the Prepar3D flight simulation software. The process by which these deliverables were created is recounted in the following paragraphs.

Before specific motors were considered, a series of feasible motor design setups were evaluated. The benefits and negatives of servo motors and induction motors were considered. Parameters including: price, footprint size, drive compatibility, operating temperature, power consumption and motor efficiency were taken into account. While servo motors were deemed the most appropriate means to operate the simulator, the budget restraints prevented a full scale integration of servo motors. To compromise, a hybrid solution using both a Moog servo motor and the simulator's original induction motors was created.

Using the mechanical team's derived force requirement, the team calculated the necessary motor specifications, sized and finally purchased a new Moog servo motor with a DS2110 drive and accompanying gearbox. Before the servo order finished processing, the electrical team went to work repairing and installing the induction motors. Available to the team were induction motors from last year's prototype model. These particular motors were slightly different models of the induction motors to be installed, and lacked braking abilities. However, their other components were in perfect working condition. The team stripped their intact housing shells and stator pieces and installed them in the braking induction motors. The gear boxes, were then connected to their respective motors. Using the mounts available on the simulator base, the motors were then mounted and adjusted as needed. Limit switches were then purchased and installed to invoke immediate system shut down should the simulator lose control or move outside of its acceptable range of motion.

The second deliverable was providing user controllability of the simulator. The goal was to have an individual control the 3 axis motion using a joystick input. This joystick control system was first prototyped using hobby servo motors from RadioShack and an Arduino microcontroller. The Arduino sketch successfully moved these motors between limiting angles set manually by the programmer. After wiring the induction motor drives and building protective power relay circuits, the code carried over with few modifications and resulted in the induction motors running as expected and responding to the joystick input. This is an open loop control, but the

serial encoder feed-out pins have been studied and their outputs documented for future closed loop positional control. The servo motor drive runs on $208V_{AC}$ and is enabled from a computer, via an RS232 cable, using a custom GUI software developed. The Arduino controller can be connected to the Servo drive's general I/O port.

The third deliverable was the integration of an F4U-4A Corsair airplane model into the Prepar3D simulation software. The model was successfully added and can be flown via standard computer commands through the software GUI. Software development files, source files, documentation and helpful functions have been collected and saved for future teams to utilize.

10.2 Project Integration Challenges

One of the biggest project challenges was the time restraints imposed upon the team by the delayed delivery time of the servo motor system. Upon delivery, software incompatibility and a number of manual discrepancies and omissions had the team frequently contacting Moog applications engineers which required a lot of footwork on behalf of the senior design team members.

The DS2110 analog drive being used was recently discontinued by Moog, which may explain the existence of manual discrepancies and software incompatibility. The first challenge was using the correct serial protocol communication between the computer and the drive. Unmentioned in any hardware specifications, it was finally determined that an RS232 cable was the incorrect communication protocol with the drive and an RS232 null modem serial connection was required. The manuals also neglected to explain how to clear the factory faults from the drive, which are in place to ensure no accidental torque is exerted while setting up. Much research was done to clear these faults and begin motor configuration.

To operate the DS2110 drive, two software packages were initially tried: WinDrive and a custom commissioning software created by Moog. WinDrive ultimately failed, and the initial software sent by Moog was only partially compatible with the discontinued drive. This software version didn't "know" all the memory locations and so could only partially communicate with the drive. This initially lead the team to believe the faults originated from human error.

Compatible, older software was later acquired from Moog. This software was compatible with the drive, but did not support the specific actuator model. This commissioning software is typically used for linear actuator models, so a linear actuator with same parameters was selected to represent the rotational actuator being used.

Once compatibility was established, the hardware and software enabling conditions proved to be another roadblock. The team connected microcontroller inputs to the drive before set-up occurred and pulled the enable pin high. This resulted in another unexpected drive fault. If the enable pin is on at drive startup, the drive automatically faults and prevents any future operation. To overcome this, the drive was restored to factory conditions and the drive reconfigured. Dynamic position and velocity faults also occurred but were cleared once time-out parameters were adjusted and the PIV gains tuned by estimation.

The last hurdle in establishing drive communication was its out of sync commutation phase. Because its commutation was out of phase, the drive communication was not reaching the

actuator at the right time and the actuator shaft would spin without user commands. This phase was re-calibrated using high level Moog access codes as provided by their application support engineers.

10.3 Future Recommendations

After meeting all of the deliverables for the 2013-2014 senior design project, the path forward for the 2014-2015 team is laid out as follows. The mechanical engineering team's goals next year will consist of four distinct parts. The first part will be taking another colleges reverse engineering efforts of the Corsair cockpit and turning the data into a useable cockpit for the simulator. The other college's work contains the SolidWorks model of the Corsair cockpit along with the tooling needed to construct a Corsair cockpit. Therefore, the team will be tasked with modifying the Corsair cockpit so that it not only attaches to the simulator, but also is light enough to be mounted to the simulator.

The second task laid out for the mechanical engineering team is the design and implementation of an interior for the Corsair cockpit. The entire inside of the cockpit must be designed to not only look like the interior of a Corsair cockpit but also function like the interior of a Corsair cockpit. Heavy research into the simulator industry and the design of the Corsair cockpit will need to be done to succeed in this aspect of the project.

The third goal for next year's mechanical engineering team is to continue to explore the possibility of implementing an outside the cabin visualization system. The system is designed by Immersive Display Inc. and a structure to secure the visualization system will need to be designed. This part of the project will involve communication with the Immersive Display Inc. to create a visualization system that is both functional and cheap.

The fourth part of the project is to finish updating the simulator motion base. The 2013-2014 team only installed one new servo motor gearbox combination due to limited funds. Once funding becomes available, two new servo motors and two new gearboxes must be purchased to finish updating the simulator base. In addition, two more motor mounts must be fabricated. However, all selection work for the motors and gearboxes has been done as well as the design of the motor mounts.

Future recommendations for future senior design teams as interpreted by the electrical team members are as follows.

The first recommendation is that a team of CompE and/or CSE students also be assigned to work on this project. They will be responsible for tasks including but not limited to software source file editing, coding motor feedback into the Arduino sketch, and generating real time log-files from Prepar3D to be read by the Arduino and control the simulator. Also note, a **permanent** project computer should be used for these tasks. The local senior design desktop computer in Castleman does not install the software properly.

Electrical engineers are still integral to this project to work on the installation and wiring of the other two servo motors, the manipulation and investigation of the fourth spin motor, the

installation of safety features and future work on the wiring, installation and IO requirements of the cockpit and control panels.

The next recommendation is the refinement of the simulator's motor control. The motors should operate using positional feedback. They should also operate with a fourth degree of motion; spin. Finally, two more servo motors should be integrated into the simulator by the culmination of the design process. It is important to note that as the servo motors get integrated into the base, the code controlling the motors is going to change drastically.

The final recommendation is that should the final cockpit be delivered to the University, the team must rewire and install the cockpit and develop adequate connectors from the cockpit to any controllers required for simulator function.

In conclusion the 2013-2014 senior design team accomplished all of the assigned deliverables for the project. The ETC Gyro IPT flight simulator is now an updated motion base with motion in the heave, pitch, and roll axes. Next year's team is tasked with finalizing the update to the motion base once funds are secured and creating a functioning cockpit for the simulator.

A1 Nomenclature

Actuating Arms – A set of three arms bolted to the primary motors of the motion base, with one actuating arm per motor.

The actuating arms are adjusted by their respective motors, affecting the pitch and roll of the cockpit.

Actuator – Linear or rotational motor used to control a system.

Cockpit – The aircraft flight deck, containing the aircraft's flight controls and monitoring systems. In the scope of this report, the cockpit will refer to the compartment of the flight simulator in which a trainee or simulator pilot would sit to experience flight simulation.

Connecticut Corsair, or **CT Corsair** – Founded by former US Air Force gunner and vintage aircraft pilot Craig McBurney, Connecticut Corsair is based out of Chester, CT, and focuses on drawing awareness to the F4U Corsair, as well as restoring a variant of the F4U Corsair – the F4U-4 – to airworthy condition [1].

Duty Cycle - maximum amount of time a motor should be continuously used for and the minimum amount of time a motor can be turned off before being turned back on.

ETC® – Environmental Tectonics Corporation. A company based out of Southampton, PA, specializing in various forms of flight and environmental training for both civilian and military markets.

ETC Gyro IPT™ – Environmental Tectonics Corporation Gyro Integrated Physiological Trainer™. A flight simulator designed and manufactured by ETC®. An ETC Gyro IPT™ donated by ETC® to Connecticut Corsair is the focus of this project and report.

Flight simulator – a device or system that artificially re-creates aircraft flight with multiple flight characteristics, including physical flight characteristics such as pitch and roll, environmental characteristics, and flight situations such as takeoff, landing, and emergencies. Flight simulators may be automatically or manually operated.

HUD – Heads-Up Display. A heads-up display projects important flight information onto a cockpit's front canopy window to allow pilots to view this information without having to look down at the cockpit's instrument panel. While modern HUD units are occasionally displayed on a pilot's helmet visor, the more common canopy application will be referred to in this report.

Instructor Panel – the large control panel which houses the simulator's external control systems, as well as multiple monitors from which an external operator or instructor can modify and monitor a trainee's flight experience.

Instrument Panel – A display of multiple gauges, sensor displays, gyroscopic displays, and other flight data displays located in front of, or near, the pilot. The instrument panel provides the pilot with all of the available information about his or her aircraft at any given time.

I/O – Stands for Input/Output. IO is communication between a processer and the world. Inputs are received data and outputs are sent data.

Lateral axis, **transverse axis,** or **pitch axis** – an axis running from the pilot's left to right in a piloted aircraft, and parallel to the wings of a winged aircraft.

Longitudinal axis, **o**r **roll axis** – an axis drawn through the body of the vehicle from nose to tail in the normal direction of flight.

Simulator Base – the platform upon which the simulator cockpit rests. All of the motion-simulating components of the simulator, such as the motors, are mounted on the motion base.

Pitch angle – designated as the angle of the simulator's nose with respect to the ground.

Prepar3D® – Flight simulation software package developed by Lockheed Martin and Microsoft®

Programmable Logic Controller -- A microcomputer typically used in precision machining applications to provide location feedback data to a CNC program. The Phase 1 project team utilized it as a simple motor control module.

Resting Position – The non-operational position of the simulator, where the cockpit is lowered to rest on its support rings.

Roll angle – designated as the angle of the simulator's sides with respect to the ground.

Scissor Arms – The diagonally-mounted arms comprising the part of the simulator's shock-absorbing system. They are split into two categories – upper scissor arms, comprising the direct link to the mounting ring for the cockpit, and the lower scissor arms, comprising the direct link to the shock springs and motion base.

Shock Springs – The springs mounted on the motion base, linked to the lower scissor arms. The shock springs provide shock-absorbing capability to the motion base and assist the actuating arms in maintaining the zero-position when raised for operation.

Variable Frequency Drive (VFD) – Used to control induction motors with varying input signal frequencies. This changes the motor shaft's output speed and torque.

Vertical axis, or **yaw axis** – an axis drawn vertically from the top of the simulator to the ground.

Vought-Chance F4U Corsair – A World War II and Korean War-era aircraft carrier-based fighter-bomber aircraft manufactured by Vought-Chance. Introduced in 1942 and easily recognized by its distinctive gull-winged design, the F4U Corsair is the official aircraft of the state of Connecticut.

Yaw angle – designated as the angle of the simulator's nose with respect to its forward-facing zero position.

Zero-position – The home position of the flight simulator while raised for operation, where $\theta = \phi = \varphi = 0$.

A2 References

1. "Abaqus CAE." *Finite Element Analysis*. N.p., n.d. Web. 15 Nov. 2013. <http://www.3ds.com/products-services/simulia/overview/>.
2. "Advanced Pilot Training." *ETC Corporate*. Environmental Tectonics Corporation, n.d. Web. 30 Sept. 2013. <http://www.etcusa.com/>.
3. "Arduino - ArduinoBoardUno." *Arduino - ArduinoBoardUno*. N.p., n.d. Web. 10 Dec. 2013. <http://arduino.cc/en/Main/arduinoBoardUno>.
4. Bartel, Charles, *Product Application Manager*, Simulation Department. "Motor Sizing Inquiry." Phone Interview. 20 Dec. 2013.
5. *Bearing Stress*. N.d. Photograph. Missouri University of Science and Technology, Web. 15 Nov 2013. <http://classes.mst.edu/civeng110/concepts/01/bearing/index.html>.
6. Boucher, Henry. "Prepar3D Inquiry and Help." Phone Interview. 4 Oct. 2013.
7. Boyd, Graham, CAPINC. "SolidWorks Course Opportunities." Chester Airport Site Visit. 29 Sept. 2013
8. Cassenti, Brice, PhD. "ANSYS Tutorial." Personal interview. 2 Oct. 2013.
9. Cassenti, Brice, PhD. "ANSYS Tutorial." *engr.uconn.edu*. N.p.. Web. 10 Oct 2013. <http://engr.uconn.edu/~cassenti/>.
10. Cassenti, Brice, PhD. "Finite Element Analysis." *engr.uconn.edu*. N.p.. Web. 10 Oct 2013. <http://engr.uconn.edu/~cassenti/>.
11. "Chance Vought F4U Corsair." *F4U Corsair History*. N.p., n.d. Web. 1 Dec. 2013. <http://www.f4ucorsair.com/history.html>.
12. Connecticut Corsair LLC. "Connecticut Corsair." *Connecticut Corsair*. N.p., n.d. Web. 20 Sept. 2013. <http://www.connecticutscorsair.com/>.
13. G. King, *Environmental Tectonics Corporation*; C. McBurney, *Connecticut Corsair*; Y. Liu, D. Synnott, M. Winczura, *University of Connecticut*; Site Meeting at University of Connecticut. 17 Nov. 2012
14. Malla, Ramesh, PhD. "Spring Constant Analysis and Approach." Personal interview. 7 Oct. 2013.
15. "McMaster-Carr." *McMaster-Carr*. McMaster-Carr, n.d. Web. 30 Sept. 2013. <http://www.mcmaster.com/>.
16. Mealy, Tom. "Project Storage and Fabrication." Site Meeting at University of Connecticut. 2 Oct. 2013.
17. *Mechanical ANSYS APDL*. Vers. 14.5. Cecil Township, Pennsylvania: ANSYS, 2013. Computer software.
18. Motaref, Sarira, PhD. "Spring Constant Analysis." E-mail interview. 6 Oct. 2013.
19. Operations & Maintenance Manual (2000). *Gyro IPT: Integrated Physiological Trainer*. Environmental Tectonics Corporation, Southampton, Pennsylvania. University of Connecticut.
20. Popoli, Bill. "Interview with IBAG." Personal interview. 20 Sept. 2013.
21. *Prepar3D.* Vers. 1.4. Bethesda, Maryland: Lockheed Martin, 2012. Computer Software.
22. Phung, Lam. *Abaqus Tutorials*. 2013. Video. www.YouTube.comWeb. 1 Nov 2013. <http://www.youtube.com/user/ngoclam250?feature=watch>.
23. "Picture of the Vought F4U-4 Corsair Aircraft." *Photos: Vought F4U-4 Corsair Aircraft Pictures*. N.p., n.d. Web. 10 Dec. 2013. <http://www.airliners.net/photo/Vought-F4U-4-Corsair/0869430/L/>.
24. Ratliff, Christian, Servo Tech Representative. "General Servo Motor Inquiry." Personal Interview. 18 Sept. 2013.
25. "Servo Tech, Inc." *Servo Tech Products*. Servo Tech, Inc., n.d. Web. 29 Sept. 2013. <http://servotechinc.com/>.
26. *SolidWorks*. Vers. 2014. Vélizy-Villacoublay, France: Dassault Systemes, 2014. Computer software.
27. "Synchronous Speed of Electrical Motors." *Synchronous Speed of Electrical Motors*. N.p., n.d. Web. 1 Dec. 2013. <http://www.engineeringtoolbox.com/synchronous-motor-frequency-speed-d_649.html>.
28. "Vought F4U Corsair." *Wikipedia*. Wikimedia Foundation, 16 Dec. 2013. Web. 12 Nov. 2013. <http://en.wikipedia.org/wiki/Vought_F4U_Corsair>.

29. "Young's Modulus." *Engineering Toolbox*. N.p.. Web. 15 Oct 2013. <http://www.engineeringtoolbox.com/young-modulus-d_417.html>
30. CA73477 SUPPORT FILES FOR 902B SOFTWARE. 1 April 2014.
31. DS2110 COMMUTATION OFFSET PROCEDURE. Moog. 10 April 2014

A3 Supplementary Analysis Data

Table 1-1:Spring Coefficient

Weight (kg)	Angle (Rad)	Displacement (m)	Spring Coefficient (N/m)
2.27	0.007	0.001	64864
4.54	0.01	0.001	92664
6.8	0.016	0.002	86489
9.07	0.017	0.002	109249
11.34	0.017	0.002	129734
13.61	0.024	0.003	111206
15.88	0.024	0.003	129741
18.14	0.026	0.003	138392
20.41	0.026	0.003	155691
22.68	0.029	0.003	157268
27.22	0.033	0.004	163897
31.75	0.035	0.004	181656
36.29	0.043	0.005	169492
40.82	0.047	0.005	173034
45.36	0.052	0.006	173049
49.9	0.054	0.006	184220
54.43	0.059	0.007	183253
58.97	0.063	0.007	187508
63.5	0.07	0.008	181767
68.04	0.07	0.008	194750
72.57	0.072	0.008	200237
77.11	0.079	0.009	196235
81.65	0.085	0.01	192816
86.18	0.089	0.01	193575
88.04	0.093	0.011	189235
92.57	0.096	0.011	192852
97.11	0.099	0.011	196601
101.65	0.105	0.012	194162
106.18	0.108	0.012	196307
108.04	0.12	0.014	179560
112.57	0.124	0.014	181854
117.11	0.128	0.015	183276
121.65	0.134	0.015	182219
126.18	0.138	0.016	183306
130.72	0.141	0.016	185237
135.25	0.145	0.016	187077
139.79	0.148	0.017	188834
144.33	0.152	0.017	190513

Table 1-2: Lift Angular Velocity

θ	α	β	Distance (cm)	Angular Velocity Of Cam(deg/s)	Cam RPM
90	163.47	73.47	36.29	534.68	89.11
80	176.4	76.4	36.86	1933.06	322.18
70	170.79	79.21	36.7	356.94	59.49
60	158.18	81.82	35.86	204.7	34.12
50	145.83	84.17	34.39	149.51	24.92
40	133.82	86.18	32.39	122.67	20.45
30	122.18	87.82	29.94	108.2	18.03
20	110.98	89.02	27.17	100.5	16.75
10	100.25	89.75	24.18	97.19	16.2
0	90	90	21.1	97.15	16.19
-10	80.25	89.75	18.01	99.92	16.65

-20	70.98	89.02	15.01	105.45	17.57
-30	62.18	87.82	12.16	113.98	19
-40	53.82	86.18	9.53	126.24	21.04
-50	45.83	84.17	7.15	143.56	23.93
-60	38.18	81.82	5.06	168.61	28.1
-70	30.79	79.21	3.28	207.03	34.5
-80	23.6	76.4	1.84	272.39	45.4
-90	16.53	73.47	0.73	408.54	68.09
-100	9.51	70.49			

Table 1-3: Roll Rate vs. Cam Angular Velocity

Angle of Platform Φ	Angle of Cam θ	Angular Velocity of Cam	RPM Cam
22.59	80	1100.69	183.45
21.86	70	487.01	81.17
20.2	60	336.29	56.05
17.79	50	271.75	45.29
14.81	40	238.43	39.74
11.41	30	220.28	36.71
7.73	20	211.05	35.17
3.89	10	208.05	34.67
0	0	210.13	35.02
-3.85	-10	202.41	33.73
-7.86	-20	248.25	41.37
-11.12	-30	247.99	41.33
-14.39	-40	275.38	45.9
-17.33	-50	315.41	52.57
-19.9	-60	375.49	62.58
-22.05	-70	471.2	78.53
-23.77	-80	641.38	106.9
-25.03	-90	1017.59	169.6

A4 Final Wiring Diagram